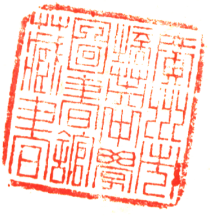
电脑学习机两周通

王福源　段黎　张海宁　编著



人民邮电出版社



电脑学习机两周通

王福源　段黎　张海宁　编著

人民邮电出版社

登记证号(京)143号

图书在版编目(CIP)数据

电脑学习机两周通/王福源等编著.-北京:人民邮电出版社,1994.12
ISBN7-115-05447-9

Ⅰ.电… Ⅱ.王… Ⅲ.电子计算机-基本知识 Ⅳ.TP3-49

内 容 提 要

随着电脑的家庭化,电脑学习机的使用越来越广泛。本书分析了电脑学习机系列在电脑家族中的位置,并介绍了它的应用范围、选购、配置、使用和程序编制方法等。本书为增强趣味性,给读者以直观的感受,非常注重运用例子。在分析、运行实例中,使读者掌握语言和编程序的技巧。所选例子本身也很具应用性。

电脑学习机两周通

王福源 段黎 张海宁 编著
责任编辑 王亚明

*

人民邮电出版社出版发行
北京市朝阳门内南竹杆胡同111号
北京密云春雷印刷厂印刷
新华书店总店科技发行所经销

*

开本:787×1092 1/32 1995年1月 第一版
印张:6.125 1995年5月 北京第2次印刷
字数:139千字 印刷:10 101—21 100册

ISBN 7-115-05447-9/TP·142

定价:6.50元

前　言

电脑这个词已经越来越多地介入我们的生活。电脑打字、电脑选曲、电脑程序控制等不胜枚举，甚至于填一个升学或求职的报名表也要用到电脑汉字代码。电脑进入家庭已成为必然的潮流。

顺应这一发展趋势，电脑学习机一枝独秀受到人们青睐。它以低廉的价格，方便的操作，独特的语言配置——F BASIC 满足人们学习电脑的基本要求。可惜目前有关资料、书籍尚不多见。本书搜集和比较了一些机型资料，提供给欲使用和正使用电脑学习机的读者参考。本书除了介绍学习卡(BS 卡，PS 卡)和功能卡的软件选配、使用外，还以例子为线索，从应用的角度出发，使读者迅速掌握计算机语言。所选例子本身也可作为应用软件使用。

电脑与其他家用电器如电视机、洗衣机不同，它不仅有选型的问题，而且存在“配置”的问题。配置与用户的功能要求有关。配置不当，不是花钱买了无用的功能，就是很快处于被更新换代的地位。电脑学习机在电脑家族中处于什么位置？各种型号机器的性能和扩充能力如何？本书想就这些方面的问题为您的决策、使用提供一些依据。

本书与某一具体牌号电脑学习机的随机使用说明书不同，目的是想提供您购机前的配置参考和购机后的使用指导，使您对电脑、语言和电脑学习机有较清晰的总体认识，帮您迅速进入电脑世界。由于作者学识有限，而目前国内生产的电脑学习机种类和牌号繁多，介绍中不免有挂一漏万之嫌，祈望得到读者和厂家的指正。

目　　录

第一章　电脑学习机的选配和应用范围……………………… 1

1.1　电脑学习机和电脑 ……………………………… 1

1.2　电脑学习机的选配和功能 ……………………… 5

第二章　学习卡的使用 ………………………………… 18

2.1　系统连接和基本功能……………………………… 18

2.2　键盘练习…………………………………………… 29

2.3　打字游戏…………………………………………… 30

2.4　计算板……………………………………………… 30

2.5　中英文编辑………………………………………… 32

2.6　五笔字型练习……………………………………… 39

2.6.1　五笔字型编码………………………………… 39

2.6.2　五笔练习功能………………………………… 55

第三章　BASIC 语言与应用 ………………………… 58

3.1　预备知识…………………………………………… 58

3.2　BASIC 程序的编写 ……………………………… 67

3.2.1　电脑计算……………………………………… 68

3.2.2　电脑辅助教育………………………………… 81

3.2.3　家政与管理…………………………………… 94

3.3　F BASIC 编程 ………………………………… 106

3.3.1　谱曲程序 …………………………………… 107

3.3.2　游戏程序 …………………………………… 114

第四章　复杂程序编写……………………………… 135

4.1 将不同的功能块结合 …………………………… 135
4.2 在原有程序中增加功能 …………………………… 138
4.3 用框图描述功能 …………………………… 142
4.4 屏幕作图 …………………………… 147
4.5 内存占用和执行时间 …………………………… 150
第五章 使用维护问答 …………………………… 162

附录1 ASCII码对照表 …………………………… 177
附录2 错误提示表 …………………………… 178
附录3 BASIC命令、语句、函数一览表 …………………………… 179
附录4 足长预测身高表 …………………………… 181
附录5 卡通代码表 …………………………… 187

第一章　电脑学习机的选配和应用范围

究竟人们能否像操作录像机一样方便地操作电脑呢？答案是肯定的。本书欲为你解开进入电脑世界之锁，帮你登堂入室，令你感到在操作电脑方面自己的能力绰绰有余。

购买家用电器的一个普遍心态是希望“一步到位”，免得刚购买不久就面临“升级换代”。那么对于想购置电脑的人来说是一步到位好还是分步到位好呢？一步到位怎样配购，分步到位又怎样配购呢？

与其它家用电器一样，同称为电脑的机器，牌号有多种多样。更不同的是购买同一种牌号，其配件有多种搭配方式，而不同的搭配方式，价格悬殊又很大。那么除了要考虑“到位方式”外，还要考虑根据购买者自己的用途来选择搭配的问题。所以你在购买之前应对电脑的工作方法有基本的了解。

1.1　电脑学习机和电脑

“电脑”是对电子计算机的俗称。电脑的运行包括两个条件，首先要有供使用者操作的“物件”，在电脑行业里我们称之为“硬件”，如显示屏、键盘等都是硬件。光是有硬件还不能得到想要的结果，类似于算盘的口诀和使用计算器时的按键顺序，电脑的运行还要由“程序”来保证。程序就是通常所说的“软件”。

我们用图 1—1 来表示上面所说硬件与软件的关系。电脑与

算盘、计算器之根本不同点还在于它的程序（软件）可以由机器（硬件）自己记住，在需要工作时，操作员给它一个命令就可以运行，而不像算盘珠要由人一个一个地拨，计算器要由人一键一键地按。用通俗的话说，就是电脑会记忆程序里规定让它做的工作，并自动地按顺序去做。

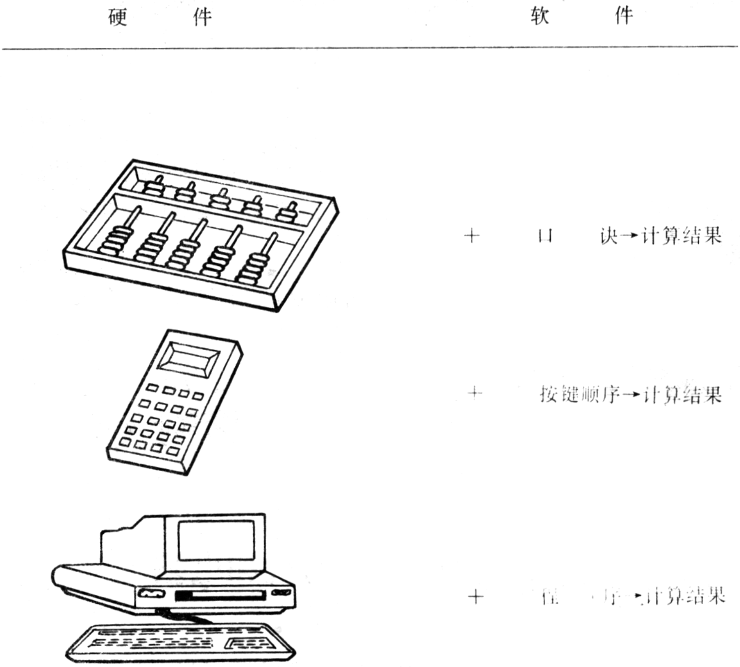


图 1—1 电脑与算盘、计算器的比较

"家用电脑"一般指的是以美国 IBM 公司的 PC 机为标准生产的电脑。把它的构成方式缩减到最小，其价格在 3000 元到 7000 元左右，可以满足普通家庭使用的需要。所以"家用电脑"一词并无严格定义。那么购买这种机器是否可以"一步到位"呢？

由于电脑技术的发展，电脑的更新速度大大快于一般家用电器，所以，就目前国内大部分消费水平来说，一步到位这一点很难做到。只能说就目前一般的家庭应用，如打字、编辑文章、打印、辅导中小学生学习、数据备查等用途来说，是一步到位了。

针对这一情况，国内一些厂商纷纷推出各种牌号的"电脑学习机"。电脑学习机的主要思想是，脱开 IBM 公司 PC 机的框框限制，设计出廉价的(通常在一千元以下)，能完成编程、打字、记忆程序等基本功能的电脑。为了达到廉价的目的，电脑学习机用家用电视机(黑白或彩色均可)作为显示器，用家用录音机作为记忆程序的工具。为吸引更多的用户，电脑学习机把游戏功能设计进去，一般市售的电视游戏卡就可以在电脑学习机上使用，达到一举几得的目的。

从以上分析可以看出，不妨先买一台电脑学习机作为学电脑的起步和过渡，先通过电脑学习机获得对于电脑配置和编程的最基本概念，或以它作为辅导儿童、青少年学习的工具。花费约 300 元至 800 元左右，打下一个基础，然后再购买档次稍高的"家用电脑"。

当然对于已经有一定电脑知识基础的读者，也可直接购买 PC 机为基础的家用电脑。PC 机也有多种档次，如 286、386 和 486，在同一档次的机器中也有多种配置方法，同样存在逐步到位和一步到位的问题，读者可参考有关资料。本书着重介绍电脑学习机的有关知识。

表 1－1 列举了 PC 机和学习机在配置方面的主要差别。由

此也可看出两者之间价格的悬殊。

从表1—1可看出，作为以PC机为主体的电脑，除了主机和打印机外，价格较高的配件是硬盘驱动器、监视器，其次是软盘驱动器。仅就硬盘驱动器来说，大容量和小容量的本身就有千元差价。就监视器来说，单色和彩色的价格差异也较大，而单色监视器虽然对大部分家庭应用可以满足，但随着技术的发展，越来越多的软件已离不开彩色显示器。就拿游戏软件来说，用单色显示器和彩色显示器的效果就大不一样。目前市面上见到的4000元左右的所谓家用电脑，一般均以单色显示器作监视器，

表1—1　　PC机和电脑学习机比较表

	家用电脑(PC机)		电脑学习机	
	配件名称	参考价(元)	配件名称	参考价(元)
主机★★	286型	1000	与键盘做成一个整体	300～900
	386型	2500		
	486型	4000		
监示器	单色显示器 彩色显示器	500 2000	黑白电视机 彩色电视机	
存放程序的工具	软盘驱动器 硬盘驱动器★	400 1200	单声道家用录音机	150
输出工具	打印机★ (选配)	1500	打印机(部分学习机可接)★	1500
汉字	汉卡★	500	已包括	
可扩充性	好		差	
耗电	200～300W		5W	

★　该项价格差异较大，本表一律按低档配置标出。

★★　主机含电源及键盘。

并且不配硬盘驱动器。而电脑学习机正是从设计上砍去了以上几项大的配件，用电视和录音机来替代，才使得价格十分低廉。

目前有些厂家已设计出带有软盘驱动器的学习机，价格在千元左右，免去了使用外接录音机的烦恼，也颇有吸引力。

电脑学习机由于价格低廉，特别适于儿童和青少年作为学习电脑知识、编程及汉字输入练习的入门工具，同时也适合于家长使用。家长可通过简单的学习，指导学生使用某些商品化的程序，也可自编一些程序辅导学生学习。因为电脑学习机的配置简单，所以对于有初中文化程度的家长，足以通过短时间的学习，掌握使用方法和编程方法。电脑学习机里配备了一种叫 F BASIC 的语言，通常叫“家庭 BASIC”，这种语言是为家庭设计的，特别易于学习和使用。通过阅读本书，您自己可以编制出一些演示性的程序。本书也给出一些 F BASIC 的实用程序。

1.2 电脑学习机的选配和功能

一、电脑学习机的构成

电脑学习机俗称“键盘机”。也就是说电脑学习机把所有的硬件都设计在键盘内了。前面已经提到，电脑学习机通常采用家用电视机作为监示器，用录音机作程序存储器。就外形而言，学习机的构成分为两种形式。一种称为合体机或称一体机，如图 1--2 所示。另一种称分体机或称组合机，如图 1—3 所示。合体机把键盘和游戏机制造在一个整体内，上面有游戏节目卡或厂家制造的专用卡的插口，两侧可插入游戏手柄(用作电脑时可以不插)，背面有电视机口、录音机口、电源口和打印机口。牌号不同，

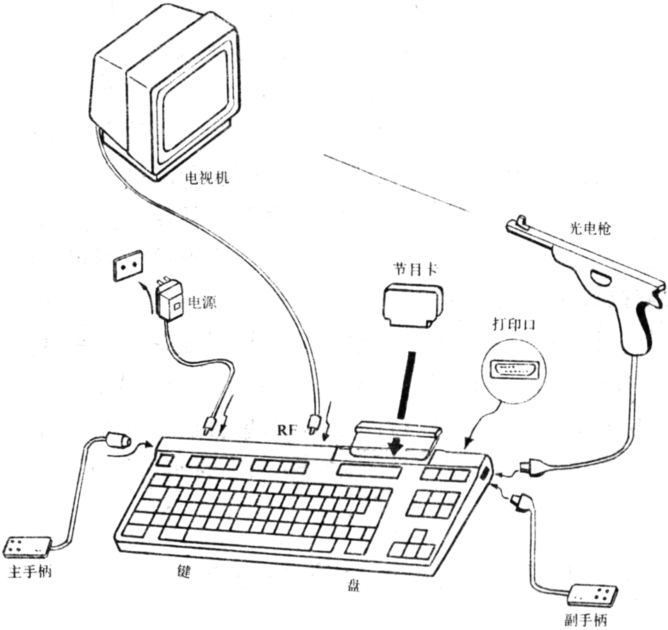


图 1—2　合体机构成

型号不同,口的多少也不同。分体机则不能独立工作,而要借助于游戏机才能工作。分体机上没有插游戏卡(或学习卡)和手柄的口子,它利用游戏机上光电枪的口子来插键盘机。

光有键盘机,电脑还不能正常工作。在购买电脑学习机时,通常会随机得到一块卡,一般称"学习卡",也有称 BS 卡、中文编辑卡等名称的。这块卡对电脑学习机用户来说非常重要。在这块卡里放进了许多厂家制造好的程序。在第二章将专门来介绍学习卡的使用。一般来说学习卡应提供下列功能:

(1) 中英文打字

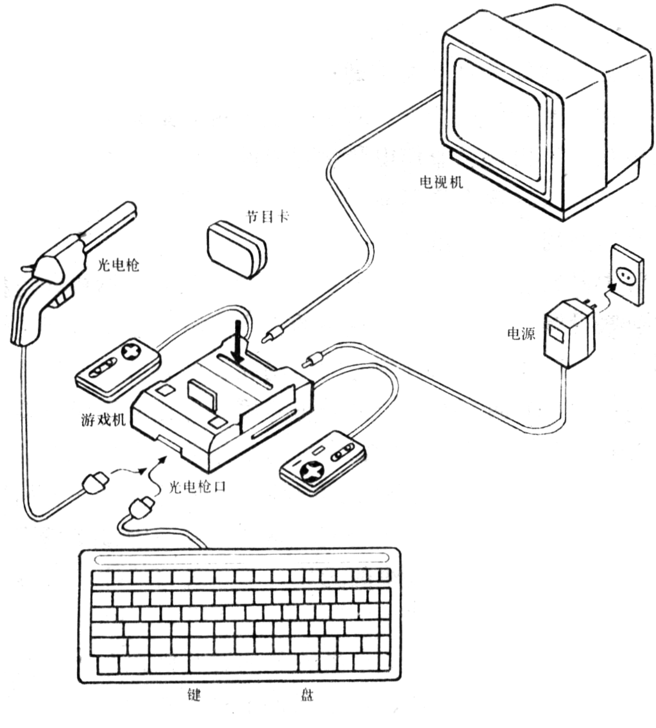


图 1—3 分体机构成

（2）计算器

（3）中英文编辑

（4）谱曲奏乐

（5）BASIC（或 F BASIC）语言，即编程序用的工具

（6）绘图

有的学习卡还提供更多的功能，如汉字五笔字型练习等等。

二、电脑学习机的选配

1. 合体机还是分体机

合体机的优点是不需借助于游戏机就可以独立工作，结构简单，使用时互相连线少。对于家庭来说，通常电视机既要作观看电视节目用，又兼作电脑学习机的监视器用。如果连线过于复杂，每次拆装十分麻烦，又无专门一块面积放置，就会影响机器的使用效率。

对于已经有了游戏机的用户，多半想购买分体机，这样可减少一些投资。充分发挥游戏机的作用。特别要注意的是国内大部分游戏机属国外某种型号的仿制品。一方面型号庞杂，牌子各异，另一方面，一些产品采用劣质元件组装而成，性能指标没有通过严格的测试。所以常有这样的情况发生，在工作十分正常的游戏机上连上电脑学习机后却不能工作，或者虽能工作，图像画面不清晰或两边有波纹出现。其原因多半是因游戏机质量太差或电源容量太小的缘故。另一种可能性是该游戏机不适于这种电脑学习机使用。

想买分体机的用户，建议带着您的游戏机和电源，在店家直接与电脑学习机连上试验，看能否正常工作。这是最简单可靠的测试方法。另外，还应注意在开启机器一段时间后，用手摸一下电源是否过热。因为键盘机是使用游戏机的电源工作的，这一额外负担有可能使电源温度升高过快而不能长时间工作。一般情况下，电源不应热到手不能摸的程度。也可平时多注意只接游戏机工作时电源的温升，在接电脑学习机后看两者温升是否相差十分明显。

对于还没有游戏机的用户，建议购买合体机，这样其电源、学习卡都是配套的，使用起来问题较少而又方便。

还有一些学习机甚至把录音功能也做在键盘上，如 GMT—92A 中文电脑学习机，其上有一个“内嵌盒式磁带存储器”。这样，录音磁带可直接放在里面使用，而不必另外再接录音机。磁带可放在机器里面，把程序“灌”到电脑学习机里直接加以运行，使用比较方便。

2. 软件的后备支援

电脑学习机的用户，除了使用学习卡上的基本功能和想在学习机上练习编程序之外，大都还想运用商品化的程序，即市面出售的软件，如中小学学习辅导程序等。如果厂家在这方面的开发能力较弱，不能供应各种实用性较强的软件，就难以充分发挥机器的效用。

就同一种牌子的机器来说，不同型号可能有不同的软件支持能力。如表 1—2 列举了裕兴家用电脑学习机（分体式）的几种型号的适用对象，其中硬件配置一栏中 8K、32K 等指标是指学习卡程序的大小，K 数越大，程序也越大。

表 1—2　　裕兴各牌号机适用对象

型　　号	硬　　件	适　用　对　象
YX3B	74 键，8K 学习卡	小学生
YX3C	74 键，32K 学习卡	中、小学生
YX4B	101 标准键盘，转换器，8K 学习卡	中、小学生及成年人入门
YX4C	101 标准键盘，转换器，32K 学习卡	中、小学生及成年人入门

有些生产电脑学习机较早的厂家已经开发出整套中、小学教学辅导软件，如 GMT 机已有上百种之多。有些新兴厂家也后来居上，开发出了目前市场急需的应用软件，如裕兴的“打字专家”和声像教学软件，可用立体声录音机播放这种磁带，在

屏幕上显示图文并茂的学习节目并有伴音。

3. 软件的提供方式

厂家以什么方式供给用户软件也是值得注意的问题。一般有三种方式，即磁带、卡和半导体芯片。对用户来说，最方便使用的是类似于游戏卡的节目软件卡，如随机的学习卡、BS卡就是这类。只要把卡插到机器上（分体机插入游戏机上，合体机插入键盘机上）就可以使用。但如果不断使用升级的软件，如辅导小学生学习，随着年级的增高，需要不断更换软件，那么每次更换新卡，成本较高，所以宁可采用录制于磁带上的软件（当然要相应购买可接录音机的学习机）。而如果只是使用一、二种软件，并不频繁更换和升级，比如主要用来作中文输入练习，那就选卡上的软件，免得每次使用之前从磁带上把程序输送到学习机内。

最近有些学习机以半导体芯片的方式向用户提供软件。半导体芯片外形如图1—4所示。

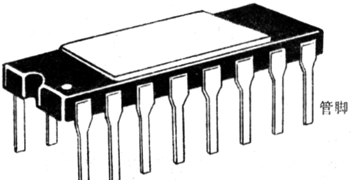


图 1—4 半导体芯片外形

只要换上一种卡上的芯片就相当于换了一个软件（如裕兴的打字专家软件）。这种方式就价格而言，介于磁带和卡之间，就实用方便来讲则与卡类似。要注意的是非专业人员在更换芯片时要搞清楚操作方法，并仔细操作，否则容易把方向搞反，也可能不当心把芯片的金属脚弄弯或折断，这样都会造成芯片失效。现在有个别学习机已经能像普通电脑一样提供软盘驱动器来存放程序。软盘驱动器的工作速度大大高于磁带，可靠性也远高于磁带并且使用十分方便，

所以颇有市场前景。可惜的是学习机的软盘尚不能与 PC 机上的软盘通用，所以程序不能交换使用。即使如此，其优越性也不是磁带录音机方式可比拟的。

4. 键盘和汉字

电脑学习机以键盘机称呼，可见键盘的重要性，因此选购时一定要注意检查键盘的质量。

质量好的键盘，“手感”很好，用手指轻轻一触即可压下，手一松即弹回原位。按下时手指明显感觉内部有触点相连。需要用力按下才有触感，屏幕上方才现出字符的键盘则质量较差。键内触点分导电橡胶和弹簧两种，导电橡胶的键，触感更为明显。而有些质量差的键盘仅靠金属导线本身的弹性和导线形成接触。这种键非但工作不可靠，而且很快会失效。

从键的安排来看，无论字母键还是数字键，普通电脑和电脑学习机的排列都同英文打字机一样，按国际通用的标准键位排列，这很适合打字的需要。但由于发操作命令用的功能键个数不一样，排列位置不尽相同，所以各种牌号的电脑学习机总键数不一样多，如有 73 键，86 键等。一般来说并不影响使用。

PC 机的键盘通常使用 101 键的标准键盘，有些厂家为配合这种键盘，生产出一种转换器，将 PC 机的 101 键盘通过转换器构成电脑学习机，便于已经有 PC 机的用户选购，也便于电脑学习机用户将来升级到 PC 机时不再需购键盘（一般键盘机与 PC 机键盘不通用）。

无论何种形式的键盘，要注意每个单键与相邻的上、下、左、右键都不可有“粘连”，即互相不能有摩擦阻碍，处于边缘位置的键与键盘边框也不应有粘连，否则会严重影响使用，特别是用键盘来练习打字，速度快时可能根本无法使用。在购买时只要逐个键试按一下即可知道。

有些键在标有数字或字母的键面上还标有一些如“田”、“口”、“早”这样的符号，这是为了方便用户在使用五笔字型打字时，输入汉字方便，详细用法在第二章会有介绍。对于想用电脑学习机来练习汉字输入的读者可考虑选购这类键盘，虽然没有这种标记的键盘也可作五笔字型输入，但初学者使用有标记键盘，会感到入门比较快捷。

前面已经提到分体机键盘和游戏机不一定能配合好，所以分体机用户购买键盘机时，最好用家里已有的游戏机去连接试验一下。

键盘上是没有汉字的，要向屏幕输入汉字必须用“代码”。例如当人们到电报局发电报时，电报局把中文电报稿逐字翻译成一串串数字，然后发给对方，每四个数字对应一个汉字，对方收到的是“电报码”的电文，电报局有电报码的对照表，可以查到汉字和码的对应关系。实际上汉字的编码方式很多，比较常用的有“汉字拼音码”、“王码（五笔字型）”等等。购机时必须弄清该学习机可以用几种编码操作，这几种码中有无适合你使用的码。少年儿童都会使用汉语拼音，如果电脑学习机用于青少年打字练习，拼音码是必备的，这样就不会产生同时要学打字又要熟悉一种码的烦恼。但拼音码的输入速度较慢，至少还要另有一种较为实用的编码，如五笔字型，这样可以在熟悉了拼音码输入，并对键盘使用有了经验之后再学习别的码。

在每一种编码方式下，电脑学习机提供一定数量的汉字供用户使用。由于汉字的数量十分巨大，不可能把所有汉字都做进学习机里。国家对汉字的使用已经有了标准，根据汉字使用的频率，即常用和不常用来分为三级。作为练习性的机器有一级到二级汉字已足够使用，不一定追求汉字的数量，因为汉字容量越大机器价格也越高。对初学者来说，比较重要的倒是看

机器是否具有比较实用的汉字练习软件，如小霸王机提供的“五笔练习”软件，裕兴机提供的“打字专家”等都是较为方便的入门工具。

5. 编程序的语言

除了使用购买的商品软件外，我们希望电脑学习机还能用于自己编写程序，这就需要解决人与电脑的交流问题。这个问题专家们早就为我们解决了，他们提供给人们“计算机语言”。用这种语言编写好程序，然后发一个命令，电脑就会按步就班地执行程序里规定要让它做的事。

计算机语言有许多种，在电脑学习机上常用的有BASIC、F BASIC、LOGO等。其中最常见的是F BASIC语言，它是为家庭设计的语言，有特别强的谱曲、绘画等功能，并且易学易记。但它也有一个缺点，就是在用F BASIC语言编写程序时不能使用汉字，这带来一些不便。如我们想编制一个猜数游戏的程序，程序开始时由电脑向操作者提示“请猜一个数”，当你从键盘输入一个数字后，电脑将答案和你输入的数进行比较，再在屏幕上告诉你“太大”或“太小”。这些提示当然可以以英文来出现，但我们更希望能用汉字来与猜数者“对话”，于是便希望有一种能使用汉字的计算机语言。

由于这种需要，一些厂家提供了另一种BASIC语言，叫“中西文BASIC”。在这种语言的支持下，可以顺利地使用汉字编程序。有的机器则干脆提供中西文F BASIC语言，使用起来更方便。

6. 打印和其他功能

大部分家用电脑学习机不提供打印功能，其原因很简单，因为打印机的价格远高于电脑学习机本身。有两种用户会希望学习机提供打印功能，一种是有可能借用或与他人合用打印机的

用户，另一种是在电脑学习机基础上有进一步发展规划的用户。后一类用户把学习机作为入门第一步，等熟悉之后再使用PC机，而打印机则可提前配置。对于有打印需要的用户，可选购有打印机接口的电脑学习机。

打印机的工作原理是以一些极细小的针，在电脑指挥下把文字或图形一点一点地“拼”出来，针越密打出来的汉字图形就越细腻。这好比我们在电视上看到的一幅画面，近看都是由一条一条的光线“拼”起来的。市面上常见的打印机有九针和二十四针两种，后者的质量远高于前者，价格也是前者的3—4倍。建议有条件的读者选购24针打印机，这对保证汉字的打印质量尤其重要。

打印机的工作是受电脑指挥的，所以购买电脑学习机时还需要清楚该机器能配接何种打印机。如裕兴的“打字专家”软件只支持EPSON牌号（或与此兼容的）9针、24针打印机；金字塔PEC—9388电脑学习机，配上该公司的“中文编辑练习卡”后，可与9针或24针打印机连通工作。

对于需要使用磁带上的程序和需要把自编程序存放在磁带上的用户，必须选购能配接录音机的电脑学习机。电脑本身虽然能记忆程序，并按程序的规定在命令下自动执行，但是一旦把机器关掉即停电后，电脑的记忆也就消失了。再次开机时，它的脑子是“一片空白”，若想运行上次用过的程序，就需要重新把预先存放在磁带上的程序再次“灌”到电脑里去。当编制一个比较长的程序，而这次只编写了一部分，下次还需接着往下编时，必须把已编好并输入电脑的这部分程序，通过录音机存储在录音带上，再次开始时，先把录音带上的程序灌入电脑，然后就可以接着往下编录，而不必每次从头开始，直到工作全部完成。大部分电脑学习机都有录音机输入（LOAD）和输出

(SAVE)口，但也有部分机型没有这类接口而不能用录音机进行程序存储，如小霸王SUBOR286型。

7. 电源

合体式电脑学习机的电源一般都随机购买，像游戏机一样，将电源与机器连通就可工作。而分体机则有的机种配电源，有的机种不配电源而借助于游戏机的电源工作，即把键盘机插头插到游戏机光电枪口上工作（不必再连电源线）。对于质量低劣的游戏机和电源会造成学习机工作不正常或不能持续长时间工作，前面已有提及，此处不再赘述。

8. 信号发射

游戏机或合体式电脑学习机通过RF线（随机购得）与电视机相连，以便把信号输送到电视机屏幕。个别质量不合格的学

表 1—3　　部分学习机性能一览表

牌　　号	小霸王286	金字塔9388	裕兴	大岛	科特	科达	GMT
结　　构	分体	合体	合体	合体	合体	合体	合体，带内嵌式磁带存储器
用户内存（K）		20	8	32	8	8	32
中西文浮点BASIC	无	有	有	无	无	无	有
中文 F BASIC	无	无	无	有	有	有	无
西文 F BASIC	有	有	有	有	有	有	无
拼音码，全拼	有	有	有	有	有	有	有
拼音码，双拼	无	无	有	无	无	无	无
五笔字型	有	有	可扩充		有		
打印口	无	有	有	有	有	无	有
打印驱动程序	无	需另配	有	有	有	无	有
按键材料	弹簧	弹簧	弹簧	橡胶	橡胶	橡胶	橡胶
录音机口	无	有	有				内嵌式磁带机

习机，信号发射能力过强，以至于影响附近的电视机正常收看，购买到这样的电脑学习机会带来许多烦恼，因此有条件的话最好购机时能用另一台不作监视器的电视机试试是否会受影响。

综上所述，电脑学习机的购置要考虑功能和质量两个方面，还要考虑暂时不用的，但可能从厂家得到的后备支持和这种支持的方便程度等几个方面。表1—3列举了几种机型的功能，读者可以参考。表中所列的某些功能后面还会解释。

三、中华学习机

上面所提供的关于电脑学习机的概念并不包括中华学习机这类电脑，但由于中华学习机曾经一度流行，目前社会上仍有相当的拥有量，而且国内仍有厂家在生产，所以在此处也略作介绍，以便读者选购时作一比较。

中华学习机是国家教委组织有关高校和计算机工厂的专家共同研制、开发的适合我国国情的电脑，价格也不高。它的设计基础是曾经在国际和国内十分流行的苹果机（APPLE II），将苹果机加以彻底改造，既可用显示器也可用电视机作为监视器，并且将汉字做在主机内，用户不用另配附件即可方便地使用汉字。它的内部存储容量大约是一般电脑学习机的8倍。它采用软盘驱动器作为程序存储，一台中华学习机可接两个软盘驱动器，使用十分方便。同时它也有录音机口，若不想配软盘驱动器，可使用录音机口存储程序或使用商品程序的磁带。

经过十多年的开发和研制，在中华学习机上已经积累了大量实用和优秀的软件，如电子表格软件VISICALC、编译软件WORDSTAR和各种教学辅助软件等数百种。可惜的是中华学习机上市之时，中国家庭还一时不能接受它的价格，而当时电脑知识的普及和应用也远未到今天的程度，上市不久又受到国

际上以苹果机为基础的机型（8位机）被PC 16位机淘汰的冲击，特别是近几年，PC机的价格一降再降，其低档配置已逐渐接近中华学习机的价格，加上虽然PC机和中华学习机都使用软盘驱动器作程序存储，但他们存储的“格式”不同，两者的软件无法交流和通用，国际上大量流入国内的新版软件无法在中华学习机上运行等诸多因素，迫使中华学习机的使用受到一定限制。

这几年正是我国电脑热升温之时，PC机已部分进入家庭，此时厂商推出比中华学习机更价廉的电脑学习机正是顺应了这一潮流。中华学习机无论在质量上和性能上都远胜于目前的电脑学习机，但缺点是价格稍微偏高，并且想向上升档时无法扩展。软盘驱动器由于和PC机格式不同，不能通用。

下面列出中华学习机的若干性能供参考。

- 结　　构：合体式，机箱内包括电源；
- 用户内存：64K；
- 程序语言：中西文BASIC、LOGO、FORTRAN等多种，F BASIC可扩充；
- 监 视 器：显示器或电视机均可；
- 编　　码：有全拼、双拼、电报码，五笔字型可扩充；
- 打 印 口：有，包括打印驱动程序；
- 录音机口：有；
- 程序存储：软盘驱动器最多两个，录音口一对。

第二章　学习卡的使用

各种牌号的电脑学习机通常都附有一块学习卡，有的称BS卡，有的称PB卡等。不管怎样称呼，在该卡中都提供电脑学习机的基本功能，所以只要掌握了学习卡的使用，就对电脑学习机的每一部分都有了了解。在此基础上要考虑厂家提供的后备软件，或者自己发展自己的程序，把学习机的功能充分加以发挥。可见学习卡是每一个初学者的入门之物。

2.1　系统连接和基本功能

一、系统连接

电脑学习机的连接比较简单，对于有游戏机使用经验的读者更无什么困难。下面以合体机为例，说明系统连接步骤。

在开始连接前做好以下准备工作：

1. 准备两个适合电视机和电脑学习机电源的电源插座；
2. 确认电脑学习机和电视机电源在关（OFF）的位置；
3. 把学习卡插到电脑学习机槽口内；
4. 将电脑学习机电源输出端插入电脑学习机电源口（ADAPT）；
5. 拔下电视机天线，将RF连线一头插入电视机天线口，另一头插入学习机RF口；

6. 把电视机音量调小。

做好以上准备工作，就可将电视机和电脑学习机电源插到插座上，然后开始下列操作：

1. 先打开电视机开关，再打开电脑学习机主机开关；

2. 调整电视机接收频率，直到对准电脑学习机信号频率，例如小霸王机用第11频道，这时屏幕画面清晰。再把音量调至合适。

现在，可以在这个“计算机系统”上工作了。

在连接和使用过程中应当注意以下几个问题：

1. 不要在开关打开（ON）的情况下插上和拔下电源，这样容易造成机器损坏。

2. 不要在开关打开（ON）的情况下，插上和拔下学习卡，如果认为学习卡插入不可靠，要重新插一下，一定要关闭电源。

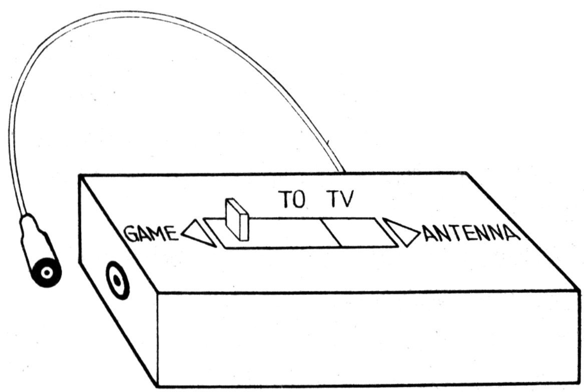


图 2—1 转换器外形

3. 有些电脑学习机的发射功率较强，只要在电脑学习机

RF 口上接一根金属线或不接线，电视机也可收到它的信号，这时只要图像清晰，也可使用，但可能影响别的电视机正常收看节目。

4. 有时候电视机可以在几个频率（频道）上收到电脑学习机的信号，比较一下画面的清晰度后选择一个最清晰的频道工作，并记住这个频道，以后不用每次调整。

5. 每次看电视时要拔掉电视机天线接口上的 RF 线很麻烦，可以购买一个“转换器”（如图 2－1 所示），把转换器的输出线接到电视机的天线口上，另外两个口，一个接电视天线，一个接学习机 RF 线。在看电视时把转换器上的开关拨到“ANTENA”处，在使用电脑学习机时则拨到“GAME”处，使用比较方便。

6. 如果需要玩游戏，在开机之前还必须接上游戏机手柄。

二、基本功能的选择

现在已经可以选择学习卡上的任意一个基本功能来使用了。每一个基本功能都代表一个厂家编写好的程序，选择了这一功能，就等于在运行这一程序。等到学习了本书后几章，可以编写自己的程序并加以运行，在此之前，先来运行厂家准备的这些程序。

各种牌号的电脑学习机，其学习卡功能和开机画面大体相仿。一般都包括下列全部或部分功能：

1. 中英文打字练习；
2. 五笔字型练习；
3. 计算板，作计算器用；
4. F BASIC，用 F BASIC 编写自己的程序；
5. 中西文 F BASIC。

通常有两种方式由用户作功能选择。一种叫做“菜单式”。好比到饭店吃饭，饭店提供一个菜单由顾客点菜。在这里是电脑提供功能菜单由用户来点功能。另一种是“询问”式，电脑不断提出一个一个的功能选项，问你要不要，如果你不要，就按某一个键（一般是 N 或 NO 两键），电脑就问下一个，直到问到你想要的那一个功能，你按下表示“要”的键（一般是 Y，YES 或 OK），这样就进入了该功能的运行。

下面分别以小霸王学习卡 V3．0 和科达为例说明菜单式和询问式两种功能选择的方法：

1．菜单式

小霸王的菜单，也就是它的开机画面如图 2－2 所示：

小霸王中英文电脑学习卡

V3．0

A．键盘练习	F．计算板
B．打字游戏	G．音乐板
C．音乐欣赏	H．留言板
D．五笔学习	I．G BASIC
E．中英文编辑	J．F BASIC

中山市小霸王电子工业公司

图 2－2　菜单式学习卡

在图 2－2 中，屏幕上方 V3．0 表示该卡的“版本号”，好比一本书出版后，经过试用发现一些问题，再作修改、再版，此时为第二版。软件也是如此，每次改版都会在功能和使用方面对老版本的缺点作出一些修改，不断完善。一般版本号越高，表

示软件越新，功能越强。如小霸王机学习卡的 V1．0 版就没有五笔字型练习和 F BASIC 两项功能。学习卡的版本号在卡的塑料壳标贴上可以看到。

现在可以任意进行“点菜”了。如想作键盘练习，则按下 A 键，想用计算器作一些计算可按下 F 键。电脑会按指定的键，进入相应的功能块。如果按下 A—J 以外的键，菜单上并无这些键的指定功能，电脑进入“玛丽医生”游戏，若此时游戏机是连着手柄的，就可用手柄进行游戏。按复位键可退出游戏，回到菜单显示。

2．询问式

此处以科达 2000 型学习机为例，说明询问式功能选择的使用。

开机后屏幕出现如图 2—3 所示的画面：

科达汉字系统

联达泽诚电子厂

A．中西文 BASIC 语言

B．F BASIC 语言

C．计算板

D．音乐板

E．留言板

进入 B～～E 请按 RESET 键

OK

■

图 2—3 询问式开机画面

从图 2—3 画面看似乎与上述菜单式无甚差别，实际上并不一样。用户并不能通过按有关的字母键进入该字母键对应的功能。例如我们想进入“C 计算板”，按屏幕提示按下：

C <CR>

（<CR> 表示 Enter 即回车键。以后我们都这样表示。）

此时出现一个双框画面，并伴有音乐，你按下任意一个键后，在双框画面上才出现：

I AM KEDA 2000 COMPUTER

即“我是科达 2000 电脑”之意，接着又显示：

NOW WE START?（我们可以开始了吗？—编者注）

接着又显示：

WHO ARE YOU?
INPUT NAME?

即“你是谁，请输入名字”之意。此时你应在键盘上打入一个名字。比如用户姓李，可键入 LI，此时屏幕下框出现你打入的“LI”，然后你再按 Enter 键，屏幕上框出现 OK！字样，然后显示：

OK!
LI IS YOU
NOW WE START

即“你好，李先生，我们开始”，然后显示：

F BASIC NEED?

下方提示为：

F1	F2	F3	F4
OK	NO	HELLO	END

如果你想要使用 F BASIC 就按 F1 键代表 OK，即“好”。我们现在想进入计算板，所以不要此功能，就按下 F2，表示“不”，屏幕显示：

CALCULATOR NEED?

即“需要计算板吗?”，此时我们可按 F1 表示愿意选择计算板。这样就进入了所选的功能。

从上述过程可以看出，询问式功能选择操作比较麻烦，一旦中间误按了键，可能要从头开始再来一遍。

通过上面讲述，我们在两种功能选择的学习卡中都可以进入所想选的功能了。在选择功能过程中我们还在键盘上做了若干操作，但实际上，我们可能对键盘上几十个乃至上百个键的使用还缺乏了解。所以我们现在来介绍一下键盘上的键以及应该怎样采用正确的指法来操作这些键。

三、键盘和指法

图 2—4 为键盘外形。尽管各牌号键盘键的个数有所不同，但排列大同小异。只要熟悉一种即可。

电脑键盘的设计和英文打字机键盘一致，电脑键盘多了一些功能键，就像我们刚才使用过的 F1、F2 等，以完成某些指定的功能，但字母和数字的位置两者都一样，所以非常适合作英文打字练习。初学者对键盘不熟悉，按键反应慢，且动作不规范，有时用单手逐个按取，这种习惯十分不好。电脑学习者应要求自己一开始就养成正确的按键指法，即使日后不想成为专



图 2—1 键盘

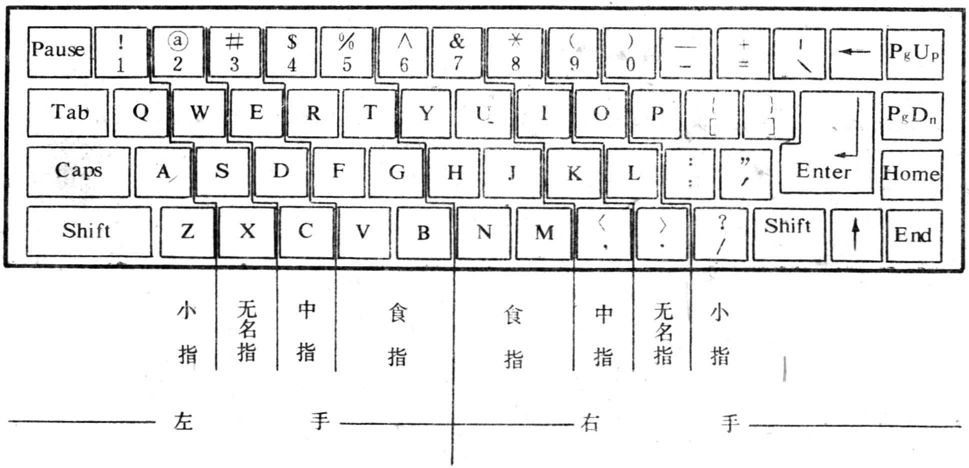


图 2—5　手指在键盘上的分工

业打字员，只是想在电脑上做些操作或输入自己编写的程序，也应该养成正确的击键习惯，并逐步达到“盲打”的程度。所谓盲打就是眼睛不看键盘而只注视文稿或监视器，仅靠手指的正确移动击键。

下面对键盘的键作一简单介绍。

键盘上的键分为字符键和功能键两类。字符键包括 26 个英文字母以及运算符号、标点符号和专用符号。功能键主要有下面几个，更多的功能键功能会在以后用到时再予以介绍。

空格键　是键盘上最长的键，位置在键盘下方中央。每按一下可使屏幕上的光标（即闪亮的方块，指出当前输入符号的位置）右移一格，如果光标右面有字符，按此键后，该字符被删除。

Enter　回车键。在书写时，为了简单常用<CR>符号表示。作为每一行或一个命令的结束。按下后光标移到下一行起始位置，准备接收新一行语句或命令。

Shift　上挡键。键盘上有某些键标记两种符号，例如，上方为左半括弧，下方为数字 8，当需要输入上方符号时，可先按住上挡键，再按下该符号键，此时屏幕出现的是“（”；如不按上挡键直接按该键，屏幕上出现的是“8”。

←　左移键。按此键可使光标在本行内左移，它也可在程序给出一系列清单让用户选择功能时作翻页。同样可解释右移、上移、下移键的作用，但上、下移键在作功能选择时不作翻页用，而是用作同一屏内容里的功能选择。

Del　删除键。按下此键光标左移，凡光标移过的字符被删除。

F1　特殊功能键。它的功能各型号机器有各自不同的规定，有些还允许使用者自己规定其功能。详细使用方法可参考

随机说明书。同样可解释 F2～F9 等键。

Break　中断键。中断程序的进行。有时用 Esc 键也可达到同样的目的。

Caps　英文字母大、小写转换键。每按一次，大写、小写变换一次。有的机器用 Caps Lock 表示。

Home　清屏幕键。清除屏幕上的所有字符。

其余更多键的使用，以后会陆续讲到。

图 2—5 画出了各手指相对应的键，在三排键中，中间一排是“基本键”。A、S、D、F 为左手基本键，I、J、L、＋为右手基本键。在击键前，手指应放在基本键位置。

利用键盘操作时应坐姿端正，身体挺直略微前倾，与键盘相距约 20 厘米，上臂自然下垂，肘靠近身体。手指放松微曲，轻放于基本键上，双脚踏地，可前后呈参差状。身体除手指外，其他部位都不要搁在键盘或桌上。击键要迅速果断，用指端垂直向键使用冲击力，迅速发力，随即收回，每次击键后手指均回到基本键位置。平时应养成眼睛注视原稿或屏幕而不看手指的习惯，就是平常说的“盲打”，这样可减少疲劳、提高速度。一般，开始时会感到不习惯，这时千万不可图快而放弃正确的操作姿式及方法，只要坚持练习，很快可达到快速、省时、省力的效果。

在了解键盘的基础上，还要通过一定的练习才能达到指法熟练。以后各节我们分别介绍一般牌号电脑学习机都包括的学习卡基本功能。为了循序渐进，我们不妨从键盘练习这一功能开始。

2.2 键盘练习

先按上面所述的方法，用菜单式或询问式进入键盘练习功能。屏幕上会出现一个“假键盘”，按一下空格键，练习就开始。

练习开始后，在屏幕的假键盘上的某一个键上出现一个红色方框（在黑白电视机上看不出颜色，但也可以练习），对应地在“真键盘”上按一下红框框住的键，如按得正确则得一分。按下键后不管正确与否，红框又会跳到别的键上，在“真键盘”上继续按红框所在的键，这样不断继续。如果红框出现在某个键位后5秒钟内还没有按下任何键，红框会自动跳到别的键上，这一次就算没有击中。

红框每跳100次为一轮练习，屏幕上键盘左上方显示的数为红框出现的次数总计，右上方则是这一轮中的得分数。例如红框跳完100次后右上方显示为87，则这一轮的击键正确率为87％。

读者在开始练习时不要太介意速度和正确率，而要把注意力放在正确的指法操作上。只要保证按指法要求练习，虽然开始时速度会慢一些，但进步会比较快。如果你觉得这种练习对你来说太快了些，那么建议你不妨自己慢慢练习一段后再来做它的练习。

按Esc键可以退出键盘练习功能，回到学习卡的菜单选择屏，以重新作功能选择。

2.3 打字游戏

打字游戏是学习卡提供给用户的另一种键盘学习程序，在小霸王等许多机型里都配有这一功能。学习本书后，读者就有能力来自己编写一个这样的程序，或许你会写得比厂家给你的更有趣和更合乎使用需要。

首先从主菜单进入“打字游戏”功能块，此时屏幕上出现0～9共十个数字，用方向键上、下移动数字旁的箭头，可使箭头指向不同的数字，数字越大，则游戏的速度也越快。初次学习的读者可选择较慢的速度进行。游戏开始后（按 Enter 键开始），从屏幕上方降下许多带有英文字母的彩球，游戏的任务是消灭这些彩球，消灭的方法很简单，只要按下与彩球的字母相同的键，彩球就被“击中”而爆炸，若按错了键，在彩球落地之前尚没把它消灭，则彩球就把屏幕下方的大地砸出一个坑，当这些坑连成一片，大地低下去一截后，这一轮游戏就告结束。

如果能通过7级或8级难度的游戏，就达到专业水平了。建议读者在具有盲打能力后再做这类游戏，否则容易片面追求速度而忽略了正确的指法。

2.4 计算板

绝大部分的电脑学习机都具备这一功能。我们前面已讲述过计算器与计算机的差别在于前者不能记忆程序，而要按照运算的目的，一个键一个键地按，必要时还要用纸记下一些中间

数据。但计算器的好处是对一些简单的问题，可省去编程序这一过程。所以这一功能对多数用户还是需要的。在学习卡里一般称为“计算板”。

进入计算板后，屏幕出现一枝羽毛笔。羽毛笔的作用相当于前面提到的“光标”，它指出当前输入字符的位置，用方向键可以控制羽毛笔的位置。

有些电脑学习机的键盘上没有乘号、除号等代数运算符号，一般在计算板上用特殊功能键来代替，如指定乘号用 F3 键代表等。不同型号的机器会对特殊功能键有不同的指定，表 2—1 列出小霸王和金字塔两种机型的指定方法，这一点不影响大家使用，因为在计算板的屏幕下方都列出指定特殊功能键的方法。

表 2—1　　计算板特殊功能键例

特殊功能键	在计算板中含义	
	小霸王	金字塔
F1	＋	＋
F2	－	－
F3	×	×
F4	÷	÷
F5	（	（
F6	）	）
F7	·	·
F8	不使用	＝

计算板还有几个与计算器使用不同之处。

（1）　在四则运算中可以使用括号；

（2）　长的算式若一行列不下，可以用下一行继续写；

（3）　算式中间不能留空格；

（4） 可以同时列多个算式，只要算式后不加等号，计算板就不出答案，直到全部算式列完，每加一处等号就出一个答案。可以利用这一功能作小学生四则运算训练和测试。

例如：列出下列算式：

8×（3＋8）－3

（16－3）×（5－4）

8×7＋7

（3＋1－2）÷（7－5）

由学生做出答案后，再让他用方向键给每个算式加上等号，这样可以校对他的答案是否正确。

至于计算板能允许的最大整数位数和最多小数位数，读者可参阅购机说明书。

2.5 中英文编辑

一、英文编辑

要完成编辑，可以进入学习卡中“中英文编辑”功能块。这时屏幕出现如下所示的菜单：

中英文编辑

D. 进入编辑

H. 帮助（命令简介）

P. 打印

X. 退出

请选择

按字母H可进入帮助命令，所谓帮助就是可以告诉你在编辑功能块里，一些常用的命令是怎样规定的，例如按F1键进入英文编辑，按F5键把光标移到文章开始等等（各机型的命令规定有所不同）。

编辑有两种方式，一种叫“插入”方式，另一种为“覆盖”方式。我们在向屏幕输入内容时，如果用插入方式的话，在光标后面的老内容仍旧存在而不被删除，适合于在有漏字符时作填补，或在老内容里插补新段落等工作。如果用覆盖方式，则输入的新内容把老内容覆盖代替，老的内容被覆盖后不再存在。

按字母D可以进入编辑。进入编辑功能后，电脑会询问你“文件名”是什么。所谓文件是一个术语，我们可以粗浅地理解成要输入的一篇文章，一封信或一个程序的名字。玩游戏的人都知道，每个游戏都有一个名字，如超级玛丽、双截龙等，名字用来区分不同的游戏程序内容。同样，我们编辑的对象也必须有一个名字，将来可以以不同的名字加以存储或运行。在中文编辑时，程序名可以用中文字，在西文编辑时，一般只允许用英文字母和数字来作名字，如ABC，A8B9等等。注意：许多机器不允许文件名的第一个字符是数字。

在有了键盘练习的基础后，要用英文编辑一个文件显然是没有什么困难的。如果你忘记了命令的规定，随时可按“帮助”来查阅。

二、中文编辑

1. 编码

中文编辑对我国大部分用户来说非常有用，可用此功能来做中文文件的输入、修改等工作。近年来我国已经开发了许多功能强大的中文编辑程序（软件），甚至于在同一页面上规定不

同的字体（如仿宋体、正楷、斜体），不同的字号，并规定一页内的行数，一行内的字数等等。

怎样才能把中文字输入到电脑中去呢？这个问题是靠“编码”解决的。所谓编码就是把每一个中文字都编上一个代码，只要在西文键盘上输入相应的代码，屏幕上就出现所想要的汉字。打个比方说，到邮电局去发电报，从北京发往上海，北京邮电局收到中文电报稿后，在一个电码本子上逐字查一下代码，每个汉字用一个四位数字代替，然后把电文译成一大串数字，将这些数字发往上海，上海的邮电局收到数字电文后，又反过来查这些数字所代表的汉字，将数字电文译成汉字电文，送交收电人。从这个例子可以看出，要用汉字代码需要几个条件：

（1） 码本：即汉字与码之间的对应关系，收方和发方都要按此约定来做才能正确使用。在电脑里我们称之为“编码”。

（2） 译员：除了有码本规定的编码外还要有译员完成正译和反译。

码本的问题已经解决，电脑中可提供用户的“码本”有多种。上面例子中谈到的电报码已经由电脑厂做在电脑中，熟悉电报码的用户可直接在键盘上用电报码输入，屏幕上会出现相应的汉字。此时电脑充当了译员。由于大部分人并不熟悉电报码，所以现在编码又有了许多别的种类，比较常用的有下列几种。

（1）拼音码

用汉语拼音来代表汉字。由于相同拼音的汉字并不是唯一的，即汉字中有许多同音字，所以汉字与拼音之间不能建立一个对应一个的关系。在输入拼音之后还要在同音字中再选择一次。一般称为“重码”，即同一个码对应多个汉字。

（2）区位码

该码把国家标准所包括的汉字分为 94 个区，每个区分为 94 个位，这样每个汉字用它的区码和位码来表示。例如“王”字，查出在 45 区的 85 位，则王的区位码为 4585。

由于区位码与汉字是一一对应的关系，所以不会发生重码。但是常用的汉字有九千多个，一般人不可能记住大量的区位码，所以输入不便，速度慢。区位码一般用来输入使用拼音码时不会发音的汉字或拼音码中不能输入的字，如“门”字，其区位码是 5671。

由于电脑学习机属于简易型电脑，所以实际上厂家并没有把全部区位汉字都做在学习机内。如小霸王机只能使用区码 54 以下的汉字，虽然如此，也已经包括了大部分常用汉字。

(3) 五笔字型

该编码是根据汉字结构和书写顺序编制的代码，其特点是输入速度快，容易实现盲打，学习起来也不困难，所以广泛流行，后面我们专门介绍五笔字型的输入方法。

初学者，特别是中、小学生对汉语拼音比较熟悉，所以建议先学习拼音码的输入。拼音码有重码，输入速度慢，但对于不经常做文字工作的用户来说，也可以应付了。如果不熟悉汉语拼音，或者虽熟悉汉语拼音，但工作涉及到大量文字处理，也可一开始就可选择五笔字型输入法。

在电脑学习机中，大部分机型是将编码功能做在学习卡中，读者在购机时须了解该机种的学习卡是否配有你所熟悉的汉字编码。一般应具有拼音和五笔字型编码。

除了上面提到的几种编码，还有一些编码，如声形码、全息码和忘形码等已陆续推向市场。

2. 拼音码的使用

电脑学习机的使用者中，有许多是中、小学生，所以用他

们熟悉的拼音码来作汉字输入的入门是很适合的。

现以小霸王机为例说明拼音码输入的操作过程。其余型号机器的操作十分类似，只要按本例操作一遍，完全可掌握输入方法。

要输入“电脑 ABC”，“电”的拼音为 dian，“脑”的拼音为 nao。首先在学习卡菜单上选择并进入编辑功能，此时屏幕出现“输入文件名：”的提示，可输入一个你选好的文件名作为本文章的文件名。设它为“D1”，按 D 和 1 两键，再按 Enter 键即可开始编辑文章。用 F2 键进入拼音方式，此时屏幕被一条红线划分为上下两部分，上部占了大半屏幕，用来显示想要输入和编辑的内容。下部仅占最后一行，用来显示你输入的拼音和对应的汉字同音字，以备进一步选择你所需要的那一个字。操作步骤如下：

按键	上部显示	下部显示	说明
d	—	拼音 d[00]	没有汉字与拼音 d 对应
i	—	拼音 di 1 堤 2 低 3 滴 4 迪 5 敌	出现多个堤的同音字，但都不是需要的，继续输拼音
a	—	拼音 dia [00]	没有汉字与拼音 dia 对应
n	—	拼音 dian 1 颠 2 掂 3 滇 4 碘 5 点	出现电的同音字，无电字
>	—	拼音 dian 1 典 2 靛 3 垫 4 电 5 佃	用>号看下一列汉字，找到电字
4	电_	拼音 dian 1 典 2 靛 3 垫 4 电 5 佃	按 4 键，取电字
n	电_	拼音 n[00]	没有汉字与 n 对应
e	电_	拼音 ne 1 呢	错按成 e
←	电_	拼音 n[00]	用←键消去错误拼音
a	电_	拼音 na 1 拿 2 哪 3 呐 4 钠 5 那	拿的同音字
o	电_	拼音 nao 1 挠 2 脑 3 恼 4 闹 5 淖	挠的同音字
2	电脑_	拼音 nao 1 挠 2 脑 3 恼 4 闹 5 淖	按 2 键，取脑字
□	电脑挠_	拼音 nao 1 挠 2 脑 3 恼 4 闹 5 淖	误碰空格键，空格键可取第一个字
←	电脑挠	拼音	用左移键把光标退到错

			字处
Del	电脑_	拼音	用清除键删除一个字
Caps	电脑_	拼音	换成大写字母
A	电脑A_	拼音	大写字母直接送到屏幕上部
B	电脑AB_	拼音	
C	电脑ABC_	拼音	
←←	电脑ABC	拼音	按5次左移键，把光标移到电字下
□…□	电脑ABC	拼音	连续按空格键，把标题放到屏幕中央位置
Enter	电脑ABC	拼音	光标跳到下一行开头，可以输入正文
Caps	电脑ABC	拼音	回到小写字母状态，接着输拼音

从上面简单的例子中可以看出，用电脑编辑文件是十分方便的事，只要经过初步的熟悉，就可运用自如。它比打字机方便之处在于既可输入中文，也可输入英文（或特殊符号），而且改错、移位、增删都十分方便，等全部完成编辑后，可以把这个文件名为“D1”的文章存在磁带上，以后需要时再从磁带上输入到电脑学习机内。

有几个问题请读者注意：

（1）长的文件一页（即屏幕上一屏画面）显示不下时，可以继续输入，前面部分虽然移出屏幕，但内容仍记忆在电脑内没有丢失，可用PgUp键和PgDn键翻页，前者是向前翻页，后者是向后翻页。

（2）上例中大写字母状态不能作拼音输入，而是直接出现在文件内容里。有些机型不采取这种操作，而是用一个特殊功能键作拼音和英文的转换。转换到拼音时，输入字母认作拼音，转换到英文时，输入内容出现在文件里。

（3）若在输入过程中需要做删、改、插入等工作而忘了应

按什么键，可随时按“帮助”功能键（上例中的F7），屏幕显示转向帮助提示，退出帮助后，原来的显示再次出现。

3．双拼

有些电脑学习机在拼音方式中还提供双拼功能（也称紧缩拼音）。我们经过一段时间练习拼音码后会发现，在汉字拼音中有许多声母和韵母需按键多次而且使用比较频繁，如：zh、ch、eng、ang、ong 等等，为了更加方便使用，我们指定某些键分别代表某个韵母或声母，这样可以大大减少击键次数，当然同时也带来一个问题，就是事先必须记住这些规定。可惜的是各种编辑功能的双拼规定不完全一样，这给使用者带来一些不便。对需要使用双拼的用户，购机时应当了解该机器是否具有双拼功能。

表 2—2 列出一种双拼的替代规定。

表 2—2　　　　双拼规定的例子

替代前	zh	ch	sh	an	ang	ao	ai	en	eng	ing	ong	u
替代后	a	i	u	j	h	k	l	f	g	y	s	v

在双拼方式下我们输入“电”字只要按 dij，输入“脑”字只要按 nk 即可。为了区别双拼与输入全部拼音方式的区别，我们有时称后者为“纯拼音方式”或叫“全拼方式”。

4．联想

有些电脑学习机提供“联想”功能。汉语的特点是大量地使用“词”，词是由若干字组成的，如“社会主义”一词由 4 个字组成。联想功能就是利用这一特点减少在汉字输入过程中的击键次数，提高效率。例如输入“游”字的拼音 you 后，屏幕下方提示能和“游”组词的字：

拼音　you　1击　2览　3戏　4行　5泳

如果需要其中某个字，只要键入对应的数字即可。如按数字3，“戏”字就出现在屏幕上方，同时屏幕下方又出现“戏”字的联想字。

联想并不是一种编码方式，而是编码使用时的手段。所以联想的使用不限于拼音码，在其他编码方式下也可使用联想。一般学习卡中设有一个“开关”，用开关来控制联想功能是否工作。如在小霸王机上，联想开关用功能键F5担任，按下F5，联想打开，提示：联想ON；再按一次F5，则联想关闭，提示：联想OFF。

2.6　五笔字型练习

学习卡中提供五笔字型练习的功能，为学习者学习和用熟五笔字型编码提供了方便。在了解这一功能的使用之前，首先要了解一下五笔字型编码规则。

2.6.1　五笔字型编码

五笔字型是我国专家王永民先生经多年研究而发明的一种编码，故此又称“王码”。这一编码专利已被国内外广泛采用。

拼音码的优点是对于会使用拼音的人来说不需要时间专门学习，基本做到上机就可以用。但一经使用，其缺点就马上暴露出来。由于存在大量的同音字，每一个输入的字都要经过二次选择才能取到，严重地影响了汉字的输入速度。所以，实际

上依靠拼音码做大量的文字处理工作是不可行的。

对一种汉字编码，人们希望它具有这样的特点：

1. 击键次数少。平均每个字需要击键的次数越少，则输入的效率越高。

2. 码短，好学易记。没有人愿意记忆大量的无规则或规则不明确的码，如上面提到的电报码和区位码，所以要求编码规则直观、易学、易记。

3. 重码少。

拼音码符合第2条要求，但由于同一个代码代表的汉字太多，用行话来说就是重码率太高，所以输入效率不高。五笔字型是满足上述条件的编码之一，大部分新版的电脑学习机都将五笔字型的学习软件装在学习卡里，供初学者使用。

与拼音码完全不同的是，五笔字型的设计基础不采用拼音的方法，而采取拼形的方法。大家知道拼音由声母和韵母相拼而成，拼形又怎样做呢？汉字的形状可谓千姿百态，怎样将它拼成呢？经过发明人多年研究，已总结了一套简明的规则，我们只要循规而行就可掌握其要义。

五笔字型编码认为一切汉字都可由“字根”相拼而成，如“李”字由“木”、“子”两个字根组成，“明”字由“日”、“月”两个字根组成。把基本字根对应到键盘上的某个英文字母，就可以拼出字来。如木对应S，子对应B，那么“李”字的编码就是SB；日对应J，月对应E，“明”字的编码就是JE等等。

我们这样说似乎把问题简单化了，读者心中会有许多疑问。首先，字根的数量一定很大，而英文字母仅有26个，是不是够用？其次，并非所有的汉字都如“李”和“明”那样简单，如“发”字怎样拼成？即使如李、明这样的字，前者为上下结构、后者为左右结构，又如何区别呢？再有，大量的字根和键的对

应关系怎样能记得住呢？同样的字根组合，是否对应单一汉字呢？比如口和八，组成“叭”还是“只”呢？

五笔字型方案已经解决了上面种种问题，只要学习和掌握这些规则并加以练习就可得心应手地使用。

五笔字型把汉字分成三个层次，即笔画、字根和字。笔画组成字根，字根组成汉字，字根按笔画分类，汉字按规则拼形。所以我们先要了解一下笔画。

一、笔画

在五笔字型中，将汉字的笔画做了科学的分类和高度的概括，并按规则 1 来规定笔画。

规则 1　笔画规则。在书写汉字时，不间断地一次连续写成的一个线条叫做笔画。笔画可分为横、竖、撇、捺、折五种。

有了规则 1，“二”、“×”、“口”均不再是笔画，并且可总结出总共只有如表 2－3 所指出的五种笔画，这就是“五笔字型”的名称由来。这为五笔字型编码奠定了基础。

表 2—3　　五种笔画

代　号	笔　画　名　称	走　　　向	笔画形状及变形
1	横	左→右	一
2	竖	上→下	丨
3	撇	左上→右下	丿
4	捺	左上→右下	丶
5	折	带转折	乙

如果用笔画直接组成汉字，理论上是可行的，也很直观，但

是致命的缺点是码长太长，如“脑”字有10画，还不算笔画多的汉字。所以迫使人们寻找比笔画还要大的组字单元，这种单元就是“字根”。

二、字根

我们对笔画已经有了规定，即规则1。由笔画组成字根时也要有个规定，否则就会乱套。例如可以把“里”选作字根，也可以把它分成“日”和“土”两个字根，也可分成“甲”和“二”两个字根。这样就不能形成拼字规则。

规则2　成形规则。一切汉字都是由基本字根组成的。

规则2中说的“基本字根”是一些组字能力很强，在常用汉字中出现次数很多的字根。五笔字型编码规则指定出130种字根为基本字根。也就是说一切汉字都用这130种字根来拼形组成。现在的问题是怎样把这130种基本字根对应到26个英文字母上，而且又能便于记忆。

要完成字根与键的对应，就用到上面讲的笔画分类。我们把这130种基本字根按起笔不同分成五种，在五笔字型中称为5个区。一区为起笔为“横”的字根如“王”；二区为起笔为“竖”的字根，如“目”；第三、四、五区分别为起笔是撇、捺、折的字根。在每个区里，又分成5个位。这样我们就得到了25个区位的字根群。即把130基本字根分成25群。每个字根群用一个代表性的字根来命名。称为“键名”。第一区第一位键名为“王”，而“王”键字根群还代表了“王”、“一”、“五”、“戋”、“龶”这些基本字根。这样做的目的，是把一个字根群对应到一个英文字母上去，如“王”键字根群对应字母G，25个字根群对应25个字母。留下一个还有别的用处，我们将在后面再介绍。25个字根群的键名如表2－4所示。

表 2—4　　键名的区位分类

区号	起笔类型	本区字根数	键名				
			1	2	3	4	5（位）
1	横	27	王	土	大	木	工
2	竖	23	目	日	口	田	山
3	撇	29	禾	白	月	人	金
4	捺	23	言	立	水	火	之
5	折	28	已	子	女	又	纟

有了表 2—4，即使还不知道键名与字母的对应关系，也可以根据区位号，写出一些汉字的数字代码了。如有些字直接是键名，“王”的代码为 11（位于 1 区 1 位），女的代码为 53 等等。又如“李”的代码为 1452，因为字根“木”在 1 区 4 位，字根子在 5 区 2 位。同样，“明”的代码为 2233，“信”的代码为 3441 等等。后面我们会看到，这样说是为了理解方便起见，并不十分严格。“信”由“人”和“言”两个字根组成，而“认”也由同样两个字根组成，所以光凭 3441 来构成代码还不够说明该汉字的全部特征。另外我们还未解决一个字母和字根群的对应关系问题。

为了解决拼形的问题，我们还是先来解决字根群怎样与字母对应的问题。

三、键盘字根

我们已经有了 25 个键名和他们率领的字根群，即有了 25 个区位。将其对应到 25 个对应字母键上，就构成了如图 2—6 所示的“字根键盘”。

在字根键盘上，每个键下方标记的字母数字含义是该字母

键代表的区位号，如14S，表示该键为S键，区号为1，位号为4，左上方"木"是在五笔字型下的键名，它率领的字根群为丁、西。同样，V键的键名为女，区位号为53，包括字根刀、九、巛、彐、臼。所有的汉字就用字根键盘上的130个基本字根拼形而成。如“着”字为U、D、H，用区位表示则是42、13、21，后面要讲到还要补充一个构字信息的码。

这么多字根排列在键上，当然不那么容易记下来，而不记下来是无法拼形的。我们可以把它总结一下，参见表2—5的字根总表。

这个字根总表上有一栏助记词来帮助记忆每个键所包括的字根，如“王”键的助记词为“王旁青头戋（兼）五一”。即王的旁边有青字的上半部分（青头），还有戋（兼）和五、一两字根。当然你还可发明一些自己认为好用的助记词。但我们不妨总结一下这么多字根布排的规律，这样就容易记住它们的位置。

规则3　排根规则。字根首笔必为区号；字根次笔一般为位号；单笔或简单笔画字根的位号等于其笔画数。

1. 首笔为区号，次笔为位号

如“戈”的首笔（第一笔）为横，所以在1区，次笔（第二笔）为折，故在5位，对应字母为A。但次笔的规律是有例外的，如“力”，首笔为折，放在5区，次笔为撇，应在3位，实际在4位，对应L。对这些例外，只有靠死记解决。

2. 单笔和简单笔画字根

单笔画一、丿、乙都在1位。二、冫都在2位（共二笔），川、巛都在3位（共三笔）。

根据规则3，再配合助记词就可记忆全部基本字根的位置。

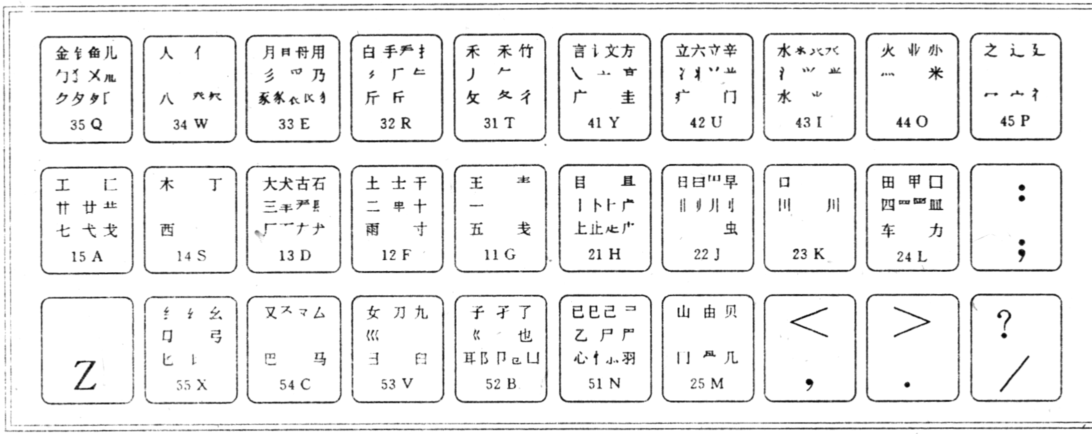

35 Q
34 W
33 E
32 R
31 T
41 Y
42 U
43 I
44 O
45 P
15 A
14 S
13 D
12 F
11 G
21 H
22 J
23 K
24 L
Z
55 X
54 C
53 V
52 B
51 N
25 M

图 2-6 字根键盘

表 2—5　　　　　　字根总表

区位	位	字母	代码	笔划	键名　基本字根	字根助记词
1 横起类	1	G	11	一	王王戋五一	王旁青头戋五一
	2	F	12	二	土士二干十甲寸雨	土士二干十寸雨
	3	D	13	三	大犬三手𡗗古石厂ナ𠂇	大犬三手（羊）古石厂
	4	S	14		木丁西	木丁西
	5	A	15		工戈开廾廿匚七弋	工戈草头右匡七
2 竖起类	1	H	21	丨	目且上止⺊卜⺊虍广	目具上止卜虎皮
	2	J	22	丨丨	日曰四早虫	日早两树与虫依
	3	K	23	川	口川	口与川，字根稀
	4	L	24	丨丨丨丨	田甲口四罒皿车力	田甲方框四车力
	5	M	25		山由贝冂⺆几	山由贝，下框几
3 撇起类	1	T	31	丿	禾竹丿𠂉彳夂	禾竹一撇双人立 反文条文共三一
	2	R	32	𠂎	白手扌手𠂉𠂉亻斤斤	白手看头三二斤
	3	E	33	彡	月⺝丹彡⺤豸乃用豕𧘇	月彡（衫）乃用家衣底
	4	W	34		人亻八癶癶	人和八，三四里
	5	Q	35		金钅勹鱼犭乂儿㇒夕夕	金勺缺点无尾鱼， 犬旁留乂一点夕，氏无七
4 捺起类	1	Y	41	丶	言讠文方广㇏亠圭	言文方广在四一 高头一捺谁人去
	2	U	42	冫	立辛冫丬丷䒑六立门疒	立辛两点六门疒
	3	I	43	氵	水氺氺水氵⺌⺍小⺌⺍	水旁兴头小倒立
	4	O	44	灬	火业灬米	火业头，四点米
	5	P	45		之辶宀冖礻	之宝盖，摘礻（示）衤（衣）
5 折起类	1	N	51	乙	已巳己コ尸尸乙心忄羽	巳半巳满不出己 左框折尸心和羽
	2	B	52	巜	子孑耳阝卩㔾了也凵	子耳了也框向上
	3	V	53	巛	女刀九臼彐巛	女刀九臼山朝西
	4	C	54		又ママ巴马厶	又巴马，丢矢矣
	5	X	55		纟幺弓匕幺	慈母无心弓和匕，幼无力

四、拆字

一个汉字要由字根来拼形，就必须把汉字拆分成字根，因此有必要对汉字的形态分析和归类，以便在编码时把分析归类的信息也编入编码里。否则，光有字根的组合还不能准确成字。

在五笔字型编码中，只要补充两个字形信息，就可以使组字有足够的“信息”。

规则 4　横竖杂规则。一切汉字可分为左右型（一型，横型）、上下型（二型、竖型）和杂合型（三型、杂型）三种组合类型。

1. 左右型。如：肚、咽、胆、侧、别、谈等。虽然“别”的左侧与“谈”的右侧分别由上、下两个字根组成，但从字的整体看是左右结构，所以归入此类。

2. 上下型。如：意、花、想等。虽然“想”的上半部分由左右两个字根组成，但从字的整体看为上下结构，故归入此类。

3. 杂合型。如团、同、乘、申、天、太等。一般来说，不明确属于左右型和上下型的可归入杂合型。

规则 5　单散连交拆分规则。基本为单；能散不连，兼顾直观；能连不交，取大优先。

1. 基本为单

有些汉字本身就是基本字根，如口、月等，就不要再加拆分，直接用相应的码键。

2. 能散不连，兼顾直观

对于散体结构我们总可以分成上下型、左右型或杂合型，所以用能散不连来控制拆分就容易形成拼形关系。规则 5 所说的能散不连，兼顾直观是指单笔画与基本字根之间的关系，而基本字根之间一般不存在连的关系。

“牛”要散拆成基本字根“𠂉”和单笔“丨”。“关”要散拆成“丷”与“大”，按取大优先，“丷”不能再拆，已经是基本字根。“生”要散拆成“丿、龶”，而拆成“𠂉、土”就不直观了。

又如“于”要拆成“一、十”，“一”和“十”两者是连的

关系，若拆成“二、丨”，虽然“二”和“丨”都是基本字根，但因为两个基本字根相交，不符合“能连不交”。“丑”字应拆分成“乙、土”，形成相连，而拆分成“刀、二”则成了相交关系。

在规则5指导下，经过自我训练，很快能掌握拆字拼形的诀窍。据王永民的统计，只需1天时间可做500汉字的拆分编码练习，这时就可以做输入练习了。

五、末笔信息

关于末笔信息的编码，在五笔字型中有一套理论指导，我们在这里只看重应用介绍，所以避开其理论，只介绍一种简单的编码记忆方法。

在五笔字型中，规定每个字最多由4个字根组成，只有对于不足4字根的汉字，才需要增加末笔信息的“识别码”。

规则6　识别码规则。4根以下加识别码。一型（左右型）补末笔字根码，二型（上下型）补两个末笔组成的字根码，三型（杂合型）补三个末笔组成的字根码。

如“沐”字拆分为“氵、木”两个字根后，判别为一型字且不足4字根，所以还要加上末笔“丶”，这样其编码为ISY，其中I、S分别为两个拆分出的基本字根，而Y则是末笔识别码。

“华”字拆分为“亻、匕、十”后，因其末笔为“丨”，“华”为二型字，不足4字根，补两个末笔字根“刂”，组成编码WXFJ，其中J为识别码。

“国”字拆分为“口、王、丶”后，因其末笔为“丶”，国为三型字，补三个末笔“氵”，组成编码LGYI，其中I为识别码。

规则7　取码规则。按书写顺序，从左到右，从上到下，从

外到内以基本字根为单位取码。

六、录入

1. 键名汉字的录入

前面已经谈到，如木、又、火这些都是键名，如果需要录入的汉字恰为键名，只要将该键连击四次，不必按空格键就完成了录入，该键名就出现在屏幕上。例如：

木　S　S　S　S

又　C　C　C　C

其它键名汉字均按此法录入。

2. 成字字根的录入

每一个键名都率领了若干字根，如表 2—5 所列。25 个区位就有 25 个字根群，这些“群众”称为“成字字根”。成字字根不全都是汉字，但有一部分为汉字。如木键的成字字根“丁”、“西”都是汉字。王键的成字字根“一、五、⺷、戋”中，“一”、“五”“戋”为汉字，“⺷”则不是汉字。

成字字根的录入可按下列顺序取码。

键名代码＋首笔代码＋次笔代码＋末笔代码

如“西”字在木键 S 上，首笔“一”代码 G，次笔“丨”代码 H，末笔“一”代码 G、所以西的编码为 SGHG。同理，“五”的编码为 GGHG，“由”的编码为 MHNG。

对于成字字根的录入，有几个注意事项：

（1）单笔画成字字根用下面公式：

键名代码＋键名代码＋L＋L

如：一　GGLL

乙　NNLL

（2）双笔画成字字根用下面公式：

键名代码＋首笔代码＋次笔代码＋空格键

如：十　FGH＋空格键

力　LTN＋空格键

八　WTY＋空格键

其中力的笔顺为㇆、丿，而我们在这里分解为丿、㇆，这与习惯写法有些不同。在五笔字型中，对成字字根刀、九、力、匕都规定用有下角伸展最远的"折"来作末笔。

（3）成字字根第一键为键名代码，发明人称之为报户口，然后依次录入三个笔画，这有利于记忆。

3. 单字的录入

单个汉字的录入用下面公式：

超过 4 字根汉字，按汉字书写顺序：第一字根代码＋第二字根代码＋第三字根代码＋末笔字根代码

不足 4 字根汉字，按汉字书写顺序：各字根代码＋识别码

这里所说的识别码按规则 6 取码。

例如，"汉"字分解字根为氵（I）、又（C），不足 4 字根，加识别码，汉字末笔为丶，字形为左右型，识别码为 Y，汉字编码为 ICY。

"副"字，分解字根为一（G）、口（K）、田（L）、刂（J），已够 4 字根，不必加识别码，也不用空格键。编码为 GKLJ。

"逾"字，分解字根为"人"（W）、一（G）、月（E）、辶（P），此处前三字根是根据书写顺序的前三个字根，第四个字根为刂，但按规则，第四个键入的应为末字根代码，所以取"辶"的代码 P，该字超过 4 字根，所以不必用识别码和空格键，

代码为 WGEP。

4. 简码输入

某些字虽然笔画和字根不十分简单，但是使用频率非常高，是常用字或称高频汉字，在五笔字型中有其压缩输入方法。

五笔字型编码把汉字的常用字分成三类，即有三个等级的简码。一级简码只要按键一次再加空格键即可完成录入。同样，二级简码和三级简码只要分别键入两个键和三个键再加空格键即可。各级简码包括的汉字总数有 3600 多字，已占据了常用汉字的绝大多数，使输入效率大为提高。简码输入提供一种简单输入方法，若你不需使用或暂时不想使用，按前面已说过的录入方法，也可完成汉字的录入。

(1) 一级简码

在英文字母 25 个键（不包括乙键）即从区位 11—55，每个键指定一个高频字，从而得到 25 个高频字的简码输入。

表 2—6　　一级简码汉字

	键名	汉字	键 2	汉	键 3	汉	键 4	汉	键 5	汉
1	G	一	F	地	D	在	S	要	A	工
2	H	上	J	是	K	中	L	国	M	同
3	T	和	R	的	I	有	W	人	Q	我
4	Y	主	U	产	E	不	O	为	P	这
5	N	民	B	了	V	发	C	以	X	厘

例如，　汉字　简码　　录入

国　L　L+空格键

以　C　C+空格键

一级简码均以两键完成录入。

(2) 二级简码

二级简码取单字代码的前两个字根简化而成。共计有 625 个。二级简码汉字只要输入它的前两个字根再加上空格键即可完成输入。

例如，“吧”字，拆分为口（K）、巴（C）两个字根，二级简码为 KC。“然”字拆分前两个字根为勺（Q）、犬（D），二级简码为 QD。

（3）三级简码

三级简码由单字的前三个字根代码加空格键组成，总共有 3300 个。对三级简码字来说，击键数采用简码和全码是一样多的，但由于减少了最后识别码的判定，所以可减少记忆、判别时间，提高输入速度。

例如，“华”字，字根拆成“人、匕、十、刂”，其全码为 WXFJ，其中 J 为识别码，三级简码为 WXF□。“情”字字根拆分为“忄、龶、月、一”，其全码为 NGEG，其中第四个码 G 为识别码，三级简码为 NGE□。

5．万能学习键“Z”

在将字根分配到英文字母键时，将字母分成 5 个区，每区内又分成 5 个位，用去 25 个键，余下一个字母 Z 尚未用到。Z 键为“万能学习键”或称“查询键”。

在编辑功能里设有一个“帮助键”，按下帮助键时可以查到在编辑功能下的许多命令提示。在五笔字型中，Z 键也可以起到类似的功能。

在字根分解过程中，我们难免会有一些一时记不清或分解不准的字根，有时也可能拿不准某字的识别码是什么，在这些情况下，我们都可以使用 Z 键得到帮助。

例如，想输入“劳”字，其字根分解为艹、冖、力。如果我们忘了“力”应用什么键，就可输入下列 4 键：

廾⼧ ZZ

此时提示行会显示出：

1. 劳 AP　2. 劳 APL　3. 蓉 APW　4. 荣 APS　5. 莹 APG

从这个提示可知道"劳"有一个二级简码 AP，另外也从劳 APL 了解到刚才拿不准的"力"的代码为 L 键。这时，只要再按下数字"1"或"2"键就可完成"劳"的录入。下次再遇到"劳"字，只要用"AP□"或"APL□"都可取到。

由上例可以看出，用 Z 键既可代替未知字根，又可查到某个字有无简码输入方式。所以它具有很强的查询功能。

作如下输入：

纟ZZZ

则提示显示为：

1. 经 X　2. 比 XX　3. 弛 XBN　4. 纺 XY　5. 纲 XM

从以上提示可以看出，"经"有一级简码，"比"和"纺"有二级简码。

又如想录入"敬"字，而记不清第二个字根的代码，这时可键入 AZKT，即第二个字根用 Z 代替，这时，提示显示为：敬 AQKT，表示代码为 A（?）KT 字根组合的字只有一个敬字。这时再按数字 1，就可录入。

如果打入 4 个 Z，电脑就会把几千个汉字逐步列出来让你挑选。

6. 词语输入

汉语的特点是大量使用词，一个编码方式如不考虑词的输入

不能算是高明的编码。五笔字型提供了词语输入的功能，并且由于它是一种拼形码，所以可以做到见形知码，使用起来很是方便。

在五笔字型方案中已选入 2300 条常用汉语词汇，如机器、计算机、艰苦奋斗、科学技术、中华人民共和国等。每个词汇只要输入 4 个字根就可全部录入。另外还给用户提供了建立新词的功能，如一位化学工作者就可能要选出诸如“氧化铁”、“聚丙烯”之类的专业词，一位作家就可能需要把作品中的人物名字造成一个词，这样可以大大提高输入速度。

（1）双字词

双字组成的词语在汉语中比重比较大。其编码方式为取第一个字的前两个字根和第二个字的前两个字根共 4 码组成。

例如， 机器：木几口口 SMKK

（2）三字词

三字词的代码为取前两个字的第一码和第二个字的前两码共四码组成。

例如， 计算机：言竹木几 YTSM

共产党：廿立小冖 AUIP

（3）四字词

四字词的代码取每个字的第一码共 4 码组成。

例如， 汉字编码：氵宀纟石 IPXD

光明日报：小日日扌 IJJR

（4）多字词

多字词的代码也由 4 键组成，其取码方法是取第一、第二、第三及最后一个字的第一码组成。

例如，中华人民共和国 口亻人口 KWWL

中国人民解放军 口口人冖 KLWP

电脑学习机属于简易型的电脑，由于存储容量和硬件方面

的种种限制，没有造词的功能，此处就不作详细介绍。至于词语录入功能，在一些早期版本的学习卡中也不具备，但随着人们对文字处理学习的需求层次提高，一些新版的学习卡中已经具有词语录入的功能。例如小霸王学习机的学习卡 V1．0 版没有词语功能而 V3．0 版则有。读者购机时可以进入五笔字型试一下，以确定所购机器是否具备这一功能。

7．单词输入例

汉　字	字　根	字　母
末	三小	DII
非	三刂三	DJD
练	纟七乙八	XANW
禺	日冂丨丶	JMHY
垂	丿一艹士	TGAF
长	丿七丶	TAY
考	土丿一乙	FTGN
书	乙乙丨丶	NNHY
得	彳日一寸	TJGF

2.6.2　五笔练习功能

较新版本的电脑学习机学习卡都装有五笔练习的软件，用以作五笔字型输入练习。下面以小霸王学习卡 V3．0 的“五笔学习”菜单选项为例，简要说明其使用方法。

进入五笔学习后屏幕出现如下菜单：

内容	字根练习	一级简码	二级简码	一级汉字	二级汉字	词组练习	速度练习
音乐 1 2 3 4 0						速度 1 2 3 4	
时间 5 10 20 30						退出 ESC	

因为它是在学习卡“母菜单”底下的五笔字型分支功能菜单，故称它为“子菜单”。在这个菜单上提供了 4 种主项功能选择，即内容、音乐、时间和速度。在内容选择里又可分为 7 个子项。主项可由上、下键控制，子项由左、右键控制，所有选择确定后，按 Enter 键可开始练习。

一、时间

这里指的是每一轮的练习时间，程序在选定时间后，作倒计时，计时结束后给出这一轮的成绩单。

二、音乐

这一子选项下的 1、2、3、4 代表 4 个不同的钢琴曲，选定后，在练习时为你伴奏。若选数字 0，则无伴奏。

三、速度

这里的速度选择只对“内容”选项中的末项“速度练习”起作用，选择不同的数字可以控制不同的出字速度，开始时可以选慢速出字，熟练之后可逐渐加快速度。

四、内容

1. 第 1～6 项练习方法

屏幕上有一群彩色气球挂着汉字从天空飘下，必须尽快派出小鸟去啄破气球，使汉字从空中掉下来。派出小鸟的办法就是输入该汉字的五笔字型码。如果选的是字根练习，只需键入相应的字根键即可。

本程序还提供了一些作查询用的功能键 F1、F2、F3、F4。按这四个键可以查出从上到下 4 个气球汉字的五笔码，词组练习则从左到右排列。这些键可以帮助初学者先通过观摩得到现场指导，但真正掌握输入还得靠自己在键盘上练习。

2. 速度练习

飞机穿过夜空，不时抛下一个汉字，地面上的火炮在巡视。尽快输入该汉字的五笔码，火炮就将汉字击落。若汉字落到地面仍未被击中，就会被沙漠吞噬。

该子项可通过“速度”选项来改变出汉字的速度，使读者循序渐进提高速度。

3. 成绩单

每次练习完成后，屏幕上发出一个成绩单，上面各项的意义如下：

时间：本次练习的选择时间；

速度：指平均打字速度，是每分钟内的正确输入汉字个数；

正确率：正确输入的字数在总字数里的百分比；

错误率：错误输入的字数在总字数里的百分比。

电脑学习机还根据练习时表现给出综合评语，低（每分钟 20 个以下），中（每分钟 20～30 个字），高（每分钟 30 个以上）。

在练习过程中，随时可按 Esc 键退出本次练习，回到五笔练习子菜单。

第三章　BASIC 语言与应用

3.1　预备知识

一、什么是 BASIC

学习电脑的目的，就是要学习如何让电脑为我们服务，也就是说，要学会指挥电脑。但是，电脑“听”不懂我们说的话，也看不懂我们写的字，怎么办呢？很简单，那就是使用一种电脑能理解的语言，这就是计算机语言。计算机语言有很多种，BASIC 就是其中的一种。与其它计算机语言相比，BASIC 更简单、易学，是为初学者设计的一种程序语言。

二、用 BASIC 指挥电脑学习机

在第一、二章里，已经学了如何进入 BASIC 功能板，现在请按上面讲述的方法进入 BASIC。如记不清过程了，请参阅有关章节。

进入 BASIC 后，屏幕上会出现一些英文字符，可不予理会，要关心的是最后一行的“OK”，以后将多次见到它，每当电脑做完一件事，它都会出现，而且总是单独占据一行，它的意思是“好了”，就是说电脑学习机处于“待命”状态，等着发号施令，如果它不出现，就无法给电脑学习机下命令。

那么怎么向电脑学习机下命令呢？那就是通过键盘。通过键盘给电脑学习机下达命令，我们称为“输入”。为了与电脑学习机的显示结果区别，凡是输入的字母或命令，本书都将在其下面划一横线。

```
PRINT    1+2  <CR>
3
OK
```

通过键盘打出来的这条命令，在屏幕上会同时显示出来，其中“<CR> ”代表“Enter”键，即前面所说的“执行键”（或回车键）。每次下达命令后，都必须按此键，告诉电脑开始执行，否则电脑不予理睬。

按下“<CR>”后，你会看到，在这条命令下面一行，出现数字“3”，这正是电脑学习机计算的结果。同时，下一行再次出现“OK”，表明可以继续用一个别的命令。

上面例子中“3”和“OK”下没有下划线，表明它们不是输入的，而是电脑显示的。

这里“PRINT 1+2”就是BASIC语言中的“一句话”，称为“命令”，即要电脑显示1+2的结果，为了加深印象，现在请输入

```
PRINT 38-26<CR>
```

可以看到结果：

```
12
OK
```

屏幕上有一个不断闪动的（白色）方块，叫作“光标”。每按一个字母或数字键，这个键的内容就出现在光标的位置上，并且光标会自动跳到下一格。光标的作用是指出当前输入的位置，实际上在学习用拼音输入汉字中已经碰到，它与在音乐板中的羽毛笔的作用类似。

三、几个常用命令

1. PRINT

请输入：

```
PRINT " A B C" <CR>
 A B C
OK
```

可以看到，命令中引号内的字母被原封不动地显示在屏幕上。再请输入：

```
PRINT "1+2" <CR>
1+2
OK
```

结果不是 3，而是 1+2，这说明 PRINT 命令的不同用法。如果 PRINT 后面的内容不带引号，则显示计算结果，带引号则按引号内原样显示。

2. “=”和变量

现在请输入：

```
A=100 <CR>
```

OK

屏幕上什么结果也没有，仍是“待命”状态，继续输入：

```
PRINT A<CR>
100
OK
```

再输入：

```
A=200<CR>
OK
PRINT A<CR>
200
OK
```

一个盒子，装进橡皮擦，它里面就是橡皮擦，装进铅笔刀，里面就是铅笔刀，这里的 A 就像一个盒子，只不过这里的 A 只能装“数”，称它为“变量”。“＝”号就相当于“装入”，等号后是要装入的内容。不能写成：

1＝A

如果这样写，电脑会告诉错了。因为它的意思是将 A 盒子装到数“1”里，当然不对了。下面几个命令更能说明问题。

```
A=1<CR>
OK
PRINT A<CR>
1
```

```
OK
A=A+1<CR>
OK
PRINT A<CR>
2
OK
```

在算术中，A＝A＋1 这种写法是错误的，而 BASIC 中就是正确的，因为这里的“＝”号不是等于，而是“装入”的意思。其实，这条命令还有个名字，叫赋值命令。让我们再来巩固一下吧，请输入：

```
A=1<CR>
OK
B=2<CR>
OK
PRINT A+B<CR>
3
OK
```

这里我们使用了两个盒子，即两个变量 A 和 B，只要它们有不同的名字，电脑会加以区别并记忆的。

在 BASIC 中规定给变量起名字的方法是：“以字母开头的字母数字串”。例如 ABC，A8，X1A 等都是正确的，而 8A（不以字母开头），A＝（＝不能使用）等都不合要求。一般还规定变量名的长度不超过 5 个字符。有些特有的字不允许作变量名，如 PRINT，详细情况可参考第五章的问答部分。

现在请输入：

```
30 A=1<CR>
40 B=2<CR>
50 PRINT A+B<CR>
```

每输入一行，虽然都按了“执行键”即“Enter”，电脑却并没有立即执行，也没有出现“OK”字样，这是什么道理呢？

BASIC 中任何一条命令都可以在它前面加上一个数，称为行号，有行号的命令与无行号的命令，其区别是，前者不能立即执行，后者可立即执行，为了区别它们，称前者为语句，而后者称为命令。如果将多个有行号的语句按行号的大小顺序排列起来，就构成了程序。

既然语句不能被立即执行，那么怎么让电脑执行呢？请输入：

```
RUN<CR>
3
OK
```

可以看到，输入该命令后，电脑执行了所有带有行号的命令，即执行了程序。这里 RUN 就是让程序执行命令。

现在回到开始提出的问题，请再输入：

```
RUN<CR>
3
OK
```

我们发现电脑又作了同样的一次计算，说明电脑“记住”了我们输入的所有语句行。也就是说输入给电脑的程序仍然存储在电

脑里。学习 BASIC 的目的，就是要最终能编写各种各样的程序。

从上述简单程序可以总结出程序的结构：

(1) 一个 BASIC 程序由若干行组成，每行内可以写一个语句，也可以写多个语句，有多个语句时，各个语句间用冒号隔开，如上面 30 行和 40 行可合并成一行：

```
30 A=1 : B=2
```

(2) 每一行都由行号、语句和结束符号“Enter”组成。行号为正整数，可以不连续。

(3) 输入完程序后，必须输入不带行号的命令 RUN，程序才能开始执行。电脑按行号从小到大的顺序一条一条地执行，直到所有语句都执行完毕。

四、与程序有关的几个命令

1. LIST

请输入清屏幕命令：

<u>CLS<CR></u>
OK

屏幕被清除干净，但我们刚才输入的程序是否丢失了呢？现在请输入：

<u>LIST<CR></u>

可以看到刚才输入的程序清单又出现在屏幕上。LIST 命令的作用就是显示程序清单，每使用一次，就显示一次。能不能只显示某几行呢？可以，请输入：

```
LIST 40<CR>                只显示行号为 40 的一行
40  B=2
OK
```

只显示行号为 40 的一行，请输入：

```
LIST 10-30<CR>
```

此时只显示行号在 10 和 30 之间的行。

LIST 命令非常有用，在修改程序时经常用到。

2. 程序的修改

前面的程序可以计算 1+2=3，若要计算 3+2=5 呢？只需将 A=1 改为 A=3 即可。修改过程为：

```
LIST 10<CR>
10 A=1
OK
```

这时就可以使用方向键“→，←，↑，↓”移动光标到要修改的地方，即“A=1”的“1”上，先按数字键“3”，然后再按“Enter”键即可。注意此时必须要按“Enter”键。否则修改不被电脑接受。这时再将光标移到原处，即 OK 下，然后再用 LIST 命令，会看到第 10 行变成了：

```
10  A=3
```

执行一下看看：

```
RUN<CR>
```

5

OK

现在请修改一下程序，改成计算3×2=6，注意，键盘上没有“×”号，而是用“*”号代替乘号。

若要去掉程序中的某一行，怎么办呢？很简单，只须输入相应行的行号并按“Enter”即可。例如去掉第20语句，则输入：

20<CR>

再用LIST就看不到第20语句了，能去掉就能添上，方法同输入程序一样，例如想在第30句与第40句之间插入一行，则可输入：

35 PRINT A<CR>
OK

用LIST查看一下就知道了。这就说明了为什么一开始行号不用1、2、3、4、5这些连续数的原因了。行号不连续是为了便于必要时在两行之间插入几行语句。

3. NEW

请输入：

NEW<CR>
OK

再输入：

LIST<CR>
OK

我们发现，原来的程序没有了，NEW 的作用就是清除程序。一般来说，在输入一个新的程序之前都要执行一次 NEW 命令。

NEW 与 CLS 的区别是，NEW 清除的是程序，而 CLS 清除的是屏幕，程序仍然存在。

3.2 BASIC 程序的编写

前面已介绍了有关电脑的基本知识，并学习了几个简单的命令。在这个基础上，我们已经可以开始学习编写 BASIC 程序了。

学习计算机语言和学习外语有类似之处，就是要多读，多模仿。在读懂别人写的程序时，无形中学到了许多方法和技巧，针对电脑学习机的特点，本节从电脑计算、辅导教学、家政与管理等方面来选用程序，目的是使读者在学习了语言之后，同时也得到一些可供实用的程序例子。读者可在这些例子的基础上，进一步练习，改进和完善这些例子，从而达到进一步提高的目的。

电脑学习机提供的 BASIC 版本有许多种，如浮点 BASIC、G BASIC、F BASIC 等等。虽然它们各有不同的特点，但其语句大部分是相同的。总的来说可以分为浮点 BASIC 和整数 BASIC 两大类。在浮点 BASIC 中，一个变量或数可以是整数或小数，例如 1/4＝0.25，并且浮点 BASIC 有函数的功能（SIN、COS 等）。而整数 BASIC 中一般没有函数，或可使用的函数很少。而且所有的数全部以整数表示，例 1/4＝0（商的整数部分为零）。G BASIC 和 F BASIC 都属于整数 BASIC 范围。

如果某种 BASIC 语言还可以允许用户在程序中使用汉字，那么我们称之为中文 BASIC，否则就称之为西文 BASIC。例如 GMT 电脑学习机内提供（不需学习卡）中西文 BASIC，中西文整数 BASIC；飞梭电脑学习机 BS 卡提供整数 BASIC，而 FS 卡提供整数 BASIC 和浮点 BASIC 供用户选择。读者在购机时要注意。

本节程序例子大部分可在任何 BASIC 下通过。读者如使用的版本与书上例子不一样，只要稍加修改即可换到自己使用的版本上去。如要把中西文浮点 BASIC 上的程序转到西文 BASIC 上去，只要把所有汉字改成英语或汉语拼音即可。

3.2.1 电脑计算

【例 3－1】 求 1＋2＋3…… … ＋100 共 100 个数的和。

实现本例计算最“笨”的办法是用一句 PRINT 语句。

```
10 PRINT 1+2+3...... ... +100<CR>
```

把所有加数都输入到 PRINT 语句中，这样做就没有利用电脑的优势。我们学习一种“循环语句”即 FOR － NEXT 语句。

程序 A：

```
NEW<CR>
5 REM CALCULATE TOTAL FROM 1-100
10 S=0                变量 S 用来装和数，S 初始为 0
20 FOR I=1 TO 100     循环变量 I 初始为 1，终值为 100，
                      每次增加 1
```

```
30 S=S+I          将 I 加到 S 里去
40 NEXT           将 I 增加 1，返回到 20 行 FOR 语句
50 PRINT S        循环结束后，打印出 S 的值
60 END            程序结束
```

在上面程序中，我们略去了每个语句行下面的下划线（表示是从键盘输入的内容）和行末的回车符，因为我们已经对此有了认识，能自行判断哪些是该输入的内容。本书以后的例子也作同样省略，除非在运行程序时，为了说明清晰起见，才加下划线和回车符。

在 20 行 FOR 语句中，我们指定了一个循环变量 I（你也可指定别的变量）。I=1 TO 100 的意思是，I 从 1 开始变化，每次增加 1，直到 100。当 I=1 时，执行 30 行把 I 和 S 之和送到 S，向下碰到 NEXT 语句时，I 增加 1 变成 2，然后返回 20 行 FOR 语句，由 FOR 语句判别 I 的值是否超过终值 100。若不超过，就继续循环，若超过就停止循环，跳到 NEXT 下面的语句 50 行执行。

从 FOR — NEXT 的执行过程我们可以看出，实际上，第 30 行语句在循环过程中被执行了 100 次，当 I 增加到 101 时，停止增加，跳出 FOR—NEXT 循环，到 50 行打印语句，把 S 的值打印出来。

循环语句充分利用了电脑“不怕麻烦”的优点，大大提高了运算效率。我们可以总结出以下几条：

（1）第 5 行 REM 可要也可不要，不影响程序。REM 后面的英文含义为“计算 1—100 的和”。这种语句在 BASIC 中称为注释语句，是写给编写程序的人自己看的，电脑并不“看”，所以电脑不在乎你写的是英文或汉语拼音。这是为了编写程序时，

编者对一些变量或程序等加说明用的。

（2）循环变量为I，当然你也可选用别的变量，如I1，ABC等，但循环内相应用到循环变量的地方也要改过来。

（3）I起始值为1，终值为100，每次增量为1，I的增量也允许不是1。例如，如果要求1—100内所有偶数的和，你只要改动20行即可：

```
20 FOR I=2 TO 100 STEP 2
```

可见初值和终值都是根据需要确定的，由STEP规定每次的增量。而如果没有写STEP，则电脑自动认为每次增加1；如果增量不是1，则要用STEP来加以确定。对于浮点BASIC，增量允许是小数，而整数BASIC，增量只能是整数。这里的增量在电脑里称为“步长”，可理解成每步循环，循环变量增加的量。

（4）每遇到一个NEXT语句，循环变量增加一个步长，然后退回FOR语句，判别是否超出终值，若未超出，则继续循环，如已超出，则循环结束转到NEXT的下面一行执行。

所以我们可以看出I和S的值按表3—1变化：

表 3—1　　　　I和S的变化

执行次数	1 2 3 4	······	99	100	101
I的值	1 2 3 4	······	99	100	101
S的值	1 3 6 10	······	4950	5050	退出循环，S不再变化

如果想验证一下，也是十分容易的，只要在循环内部加一行语句把I和S的值都打印出来：

```
35 PRINT I，S
```

我们对程序A稍作修改，即可改编成一个任意两个整数之

间的求和程序。

程序 B：

```
NEW
10 S=0                    S装总和，初值为零
20 INPUT   A              从键盘输入求和区下限
30 INPUT B                从键盘输入求和区上限
40 FOR I=A TO B
50 S=S+I
60 NEXT
70 PRINT " S="; S
80 END
```

在程序 B 中，第 20、30 两行中的 INPUT 语句是我们未见到过的。INPUT 的功能就是给变量送数。我们前面学习过“赋值语句”，用“=”给变量送数，但这只能限于在程序中给变量送数。现在我们想把程序编写成“任意上限和下限”之间的和，这样就希望使用程序的用户从键盘上将求和区间的上限和下限输入电脑。而 INPUT 语句正是这样用的。

当程序执行到 20 行时，读者可在屏幕上看到出现一个?号，就是提醒用户，该输入数了。这时将 A 的数键入，再按 Enter，屏幕上又会出现一个？号，这是因 30 行中的 INPUT 语句又向键盘要变量 B 的数据，此时再向电脑输入 B 的值，程序又依照行号向下执行。

录入无误后，可试运行一下。

```
RUN<CR>
? 1<CR>
```

```
? 100<CR>
S=5050
OK
```

若还有别的数需要计算可再次运行：

```
RUN<CR>
? 18<CR>
? 47<CR>
S=975
OK
```

你也可以事先手算出两个正确的结果，用来验证程序是否正确。

我们看到，如果要从键盘输入的数据很多时，光从键盘上出现一个问号，可能会搞不清到底轮到该输入哪个变量了。所以在 BASIC 里，可以在 INPUT 语句中，写入一些提示。

现在我们把本例改动一下。

```
20  INPUT  "A=";A
30  INPUT  "B=";B
```

其余各行完全不变。再试运行一下。

```
RUN<CR>
A=?  1<CR>
B=?  100<CR>
S=5050
OK
```

可以看出，有提示的 INPUT 语句，使用比较方便。特别是当你编制的程序要让他人使用时，提示变得不可缺少。

BASIC 规定提示部分要用分号与变量隔开。也可以把两个变量放在一个 INPUT 语句中，改成：

```
20  INPUT  "A=?  B=?"
```

第 30 句取消，运行情况为

```
RUN<CR>
A=?  B=?  1, 100<CR>
S=5050
OK
```

注意在输入两个变量的值时，中间要用逗号分开。

（5）程序末行的 END 语句，表示程序结束，称为结束语句。

【例 3－2】 打印九九乘法表

九九乘法表先要做 1×1、1×2、1×3．．．．．．1×9。然后被乘数变为 2，做 2×1、2×2、2×3．．．．．．2×9。如此变化，一直到被乘数变为 9，乘数从 1 至 9 为至。可以看出，变化的量有二个，即乘数与被乘数，被乘数每变化一次，乘数要变化 9 次。我们可以用一个“二重循环”很方便地把这一规律加以实现。

程序：

```
NEW
5  REM  9*9  TABLE
10  FOR  A=1  TO  9
20  FOR  B=1  TO  9
```

```
30  C=A*B
40  PRINT  A;"*";B;"=";C
50  NEXT
60  NEXT
70  END
```

上述程序中语句 10 和 60 的 FOR — NEXT 对称为“外循环”。语句 20 至 50 的 FOR — NEXT 对，称为“内循环”。外循环变量 A 每变化一个步长（无 STEP 语句，步长为 1），内循环变量将从初值到终值全部循环一遍。然后进行下一轮外循环。直至外循环变量超出终值。

我们将这种循环称为循环的“嵌套”，在需要的时候，循环嵌套可以是多套的，而不限于二套，其嵌套数的限制可在你所用的 BASIC 版本说明书中查到。但在一般应用中，二套和三套已完全能应付了。嵌套循环的方式可以理解成时针和分针的关系，时钟每走一格（一个步长），分针要转 60 个圈，待时针走完一圈时，分针已转了 12×60=720 圈了。所以在本例的程序中，30 行语句 C=A×B 实际被执行的次数为 9×9=81 次。同样，40 行打印语句也执行了 81 次，共打印出 81 个式子。可以试运行一下，看看结果。

```
RUN<CR>
1*1=1
1*2=2
1*3=3
   :
   :
   :
```

```
1 * 9=9
2 * 1=2
2 * 2=4
   :
   :
   :
9 * 8=72
9 * 9=81
OK
```

这里有几点要加以说明：

（1）BASIC 语言中乘法号用“*”，而不用“×”，其他运算符如下表所示。

数学符号	例	BASIC 表示	
乘×	2×3	*	2 * 3
除÷	100÷2	100/2	100/2
乘方	4^2	* *	4 * * 2 或 4 ↑ 2
（　）	8÷（1+3）	（ ）	8/（1+3）

注：BASIC 中无【　】号和｛　｝，一律用（　）表示。

（2）多重循环的 FOR — NEXT 配对。应以最里层的 FOR 开始，向下寻找第一个遇到的 NEXT 称为一对，然后向外退一层按上法配对，直到所有的 FOR 都对上对应的 NEXT。如果 FOR 和 NEXT 不配对，个数不符，电脑将无法执行程序，提示语法有错。

（3）PRINT 语句中分号的作用为当几个要打印的内容（包括变量和字符串）中间用分号隔开时，相当于通知电脑，在打印时，中间不要留空，打完一项内容，紧接着打印下一项内容，

如最后一项打完后不加分号，则碰到下一个 PRINT 语句时，自动换一行打印。如果最后一项内容也加分号时，碰到下一个 PRINT 语句时，仍接着上一个 PRINT 的最后一项内容后面打印。

本例 40 行 C 后面没加分号，所以每次换行打印下个算式。

【例 3－3】 编一个星期推算程序，要求从键盘上输入被推算的年、月、日后，电脑能告诉该日期是星期几。假设已知公元前 1 年 12 月 31 日为星期日。

我们要计算某年某月某日是星期几，就要算出该日离公元前 1 年 12 月 31 日有多少天，然后将天数除以 7（一周 7 天），余下的数就是星期数，如余数为零，则是星期天。

程序（整数 BASIC）

```
10  DIM  A(12)
20  FOR  I=1  TO  12
30  READ  A(I)
40  NEXT  I
50  INPUT  "NIAN, YUE, RI"; N, Y, R
60  M=N-N/100*100
70  W=M+M/4+A (Y) +R
80  S=W-W/7*7
90  PRINT "XING  QI="; S
100  DATA  0, 3, 3, 6, 1, 4, 6, 2, 5, 0, 3, 5
110  END
```

这个程序看上去写得很简单，但要读懂它还得理解其中一些技巧。这些技巧对我们日后自己编程是有用的，可以提高我们的编程能力。

首先我们遇到两个新的语句，DIM 语句和 READ — DATA 语句。

DIM 语句称为数组定义语句，所谓数组，简单地讲，就是用一个变量名来装一组数，而不是像我们前面见到的那样，一个变量只装一个数。为了区别这一组数里的各个数，用变量后加括号，括号内写上一个标号表示在这组数内的序号。这一组数就称为数组。本例中 A 是一个数组。

数组为编程序提供了有力的工具。我们经常会遇到一组性质相同的数，例如要统计一个班的成绩，有 50 名学生，每位学生有 4 门课，政治、语文、数学和物理，那么共有 4×50＝200 个成绩，如果不用数组，我们要用 200 个变量名来表示，而用数组，可分别用 4 个数组分别代表四门课，每个数组规定 50 个数，就可以方便地输入或打印其中任意一个成绩。例如用 P 代表政治分数，P 的大小规定为 P（50），则 P（1），P（2）…… P（50）分别代表 50 名学生中第一、第二……第五十名学生的政治成绩。

DIM 语句就是用来规定数组大小的语句，DIM A（12）就是通知电脑，本程序中将用到一个名叫 A 的数组，最多装 12 个数。我们称为“数组定义语句”，关于 DIM 语句的使用在 BASIC 中有几条规定：

（1）DIM 语句必须写在程序头部。你可以理解成“预订房间”，我们外出旅游之前，可用电话通知对方旅店预留若干房间。在这里我们必须预先通知电脑，我在程序中要用几个数组，每个数组最多装几个数，这样电脑就可以为你预留出“房间”，或我们前面说的“盒子”，不被别的变量占用。

（2）一个 DIM 语句中可以定义多个数组，用逗号分开，如：

```
DIM   A(12), B(30)
```

就定义了 A 和 B 两个数组，最大个数分别是 12 和 30。

（3）数组的起名方法与单个变量一样。

READ － DATA 语句是用来给变量送数，即赋值的语句。我们已经使用过 INPUT 语句给变量送数，即从键盘上给某个变量赋值。当然在程序里也可用“＝”来赋值。当要送数的数量很大时，用上面两种方法效率都不高，所以在 BASIC 里又设置了 READ 语句。READ 语句称为“读语句”，读的目标为 DATA 语句，例：

```
10  READ  A, B, C
        :
        :
        :
20  DATA  12, 8, 7
```

在执行第 10 行时电脑自动到 DATA 语句中去按顺序读数，把 DATA 语句中的 12 赋给变量 A，8 赋给变量赋给变量 B，7 赋给变量 C。可见 READ 与 DATA 是配对出现的。

本例中，READ A (I) 出现在循环语句中，I 的变化从 1 到 12，所以 I＝1 时把 DATA 中的第一个数据 0 送到数组变量 A（1）中去。I＝2 时，把第二个数据 3 送到 A（2）中去等等。到循环结束时，整个数组的 12 个单元中都装了相应的值，例 A（4）＝6，A（12）＝5。

从上面例子我们也可看出，如果没有数组，我们要将 12 个数送给相应的变量，程序会比较长而无序。

现在我们回到本例的星期推算问题本身来分析。我们的目的就是计算出第 50 行输入的年、月、日，离开公元元年 12 月

31 日有多少天，将天数除以 7，余数就是星期数。至于其间有多少个 7 的倍数，是不用关心的。

根据历法，从公元第一天开始，每过 100 年，其天数正好是 7 的整数倍。所以我们可以把输入年份中 100 的整倍数部分扣除不予计算，免得数字过大。例如，要计算 1992 年 8 月 7 日是星期几，根据第 40 行的变量规定，此时 N＝1992，我们从 1992 中扣除 19 个 100 即 1900，余下只算 92 年 8 个月零 7 天。第 60 句语句即作“扣除”计算用。

在 60 句中，N 为输入的年数，现在是 1992。N/100＊100 这一项表面看起来似乎除以 100 又乘上 100，（/即算术中的÷），等于原来的数一样。实际上并非如此。我们前面已讲过。整数 BASIC 只有整数运算，所以 N 除以 100 的结果，就是缩小了 100 倍并把后二位小数丢弃。如 1992/100 结果为 19，再将 19 乘以 100 得 1900，所以 N－N/100×100 总的效果就是把输入年份的后两位整数数值送到变量 M 中去。此时变量 M 的值为 92。

第 70 行中的变量 W 就是算出来的剩余天数。

W 由 4 项组成。

（1）第一项为 M。我们知道，通常情况，一年有 365 天，如果一年有 364 天，则正好是 7 的整倍数 364＝7×52，所以，实际上我们仍不必把每年的天数都算上去，凡是 7 的倍数部分都不必算。既然 364 是 7 的整倍数，则每年多出来 1 天要单独去计算。现在我们要计算 92 年多，那么就要加进 92。所以这一项是每年有一天“零头”的值。

（2）第二项 M/4。因为每 4 年有一个闰年，所以这一项是加入由闰年多出来的天数。

（3）第三项 A（Y）。Y 是指输入的月份。我们从 DATA 语

句中可以看出，如果是一月份，要加入A（1）＝0。如果是2月份，要加入A（2）＝3等等。这一项的由来是在计算Y个月的总天数时，还是把Y个月的总天数中，7的整倍数天数扣除，不予加入到W中去。如果Y＝1，即我们要算1月R号的天数，既然1月份未过完，所以不加数（加个零）。如果Y＝2，说明1月份已过完，而2月份未过完（只过了R天）。而一月份是31天，除去7的整倍数28天，余下则是A（2）＝3的值，仍需加入W进行计算。A（I）中其他值也是按同样方法事先算好，并写入DATA语句的。

（4）第四项R：R代表待查的日子。把它作为天数加入W中。现在要查7日是星期几，R就为7。当然你也可采用上面的做法，把R中7的整倍数扣除，将余数加入至W中，这样70句就变成

```
70  W=M+M/4+A (Y) +R-R/7*7
```

但是因为R的值不可能超过31，本身不是很大，所以这里就没有这样做，而是直接把R加到W中了。

第80行的任务是取出W扣除7的整倍数后的余数。与上面一样，在整数BASIC中W/7会把7的最大倍数求出来而把小数部分丢弃，再乘以7后，就等于W中含有7的最大整倍数，所以S就表示W除以7的余数，S也同时代表该日期的星期数。

从这个例子，我们可以看出，在编程序之前，我们要将问题先加以分析，看看要达到的目的是什么。目的搞清楚之后，再利用一些程序技巧，就可以使程序大为简化。

运行结果如下：

RUN<CR>

```
NIAN, YUE, RI?  1992, 8, 7<CR>
XING  QI=5
```

电脑告诉你1992年8月7日是星期五。

如果在浮点BASIC上运行，本程序会得不到正确的结果。原因是浮点BASIC有小数运算，所以60、70、80行中的除法运算均会带着小数出现，此时只要将这三行稍加改动即可使用。

```
60  M=N-INT (N/100) *100
70  W=M-INT (M/4) +A (Y) +E
80  S=W-INT (W/7) *7
```

详细分析可参考有关BASIC的说明。若你使用的是中西文BASIC就可以将第50行的输入提示和第90行的打印提示都改成中文，这样你的程序使用起来就更加方便了。如果你有兴趣，还可以在这个程序上给出星期几的答案后，再加上一段卡通活动的语句，或奏乐的语句，就更有趣了，关于卡通和奏乐，我们在下节介绍。第四章例4—1介绍了一个与卡通活动结合的程序。

3.2.2 电脑辅助教育

电脑辅助教育也称计算机辅助教育，这类程序的特点是以电脑为辅助手段，通过人与电脑的“对话”，或称人机交流进行生动形象的教育。通过辅助教育可大大提高学习者的积极性和参与性，并可达到寓教于乐的目的。

实际上我们在电脑学习机的学习卡中已经接触和使用了这

样的程序，例如“打字练习”、“五笔练习”等，均属于这一类。现在我们通过以上语句的学习，已经具备条件，可自己来编出一些辅助教育的程序，以供青少年、儿童或自己使用。

【例 3－4】 为小学生编写一位数乘法练习程序。以 10 道题目为一组，若答对则提示“VERY GOOD”（很好），接着显示下一道题。若答错，则提示：“SORRY，WRONG！TRY NEXT！”（错了，试下一道。）在十道题目之后给出答题正确率。

我们有了上面程序编写基础，可以把程序设计得“花俏”一些。使它具有下列功能：

（1）每 10 道题为一轮，每轮结束时给出本轮的正确率并打出所有错题式子和错误答案，以便用户分析原因，针对改进。

（2）每道题目不论答对还是答错，都给出对错的提示。

（3）每轮结束，询问用户是否还要进行下一轮，如要，则再开始新的一轮，否则就退出程序的运行。

程序（整数或浮点 BASIC）：

10 CLS	清屏幕
20 DIM A（10），B（10）	A，B 数组分别装乘数与被乘数
30 DIM C（10），R（10）	C 数组装用户答案（积），R 数组装错题序号
40 T＝0：L＝1	T 记录答对的题，L 记录答错的题目
50 FOR M＝1 TO 10	每一轮为 10 题
60 A（M）＝RND（10）	产生任意数作为乘数和被乘数，范围为 1－10
70 B（M）＝RND（10）	
80 X＝A（M）＊B（M）	X 为正确答案

90 PRINT A (M); “X”; B (M); “=”	列算式，提示用户作答
100 INPUT C (M)	用户输入的答案送 C 数组
110 IF C (M) =X THEN GOTO 160	如答对则转向 160 行
120 PRINT “错了”	否则给出错题信息
130 R (L) =M	把错题目号顺序装在 R 数组
140 L=L+1	错题号的序号增加 1
150 GOTO 180	跳过答对部分处理程序
160 T=T+1	答对时,T 增加 1,记录对题总数
170 PRINT “很好，请继续!”	给出答对提示
180 NEXT	
190 PRINT “正确率=”; T * 10; “%”	T 乘 10 为对题的百分率
200 IF L< >1 THEN PRINT “错误如下:”	L≠1 说明有错题印出提示
210 FOR M=1 TO L-1	共错 L-1 道
220 IF L=1 THEN M=L-1: GOTO 250	如没有错，这个循环不用执行
230 Y=R (M)	把错题序号送到 Y
240 PRINT A(Y);“X”;B(Y);“=”;C(Y);“?”	把错题目打印出来
250 NEXT	
260 PRINT “再试一次?(Y/N)”	下一轮提示
270 INPUT P$	输入用户意向
280 IF P$= “Y” THEN GOTO 40	如要下一轮，从头开始
290 PRINT “再见!”	否则结束程序
300 END	

本程序所有提示我们都采用中文，可在中西文整数或浮点BASIC下运行。如你只有西文BASIC，只要将提示中的中文改成英语或汉语拼音，其它不必改动，即可使用。

在110、150等行内我们使用了GOTO语句，这个语句较容易理解，即“无条件转向... ...”，如150句的GOTO 180，即命令电脑执行完140行后直接跳到180行去执行，所以称为“无条件转向语句”。

第110句中有IF － THEN语句，称为条件语句。该语句的作用是根据条件成立与否，确定执行THEN后面的语句还是跳过THEN后面的语句去执行本句的下一句。IF与THEN之间的部分为条件。现在我们的条件是C（M）＝X，如果C（M）＝X成立，即条件满足，我们称条件为真，就执行THEN后的内容（转向160行），而如果C（M）≠X，则条件为假，就从120行继续执行。

在BASIC中允许THEN后面有多个语句，中间用冒号分开。这些跟在THEN后面的语句都属于在条件满足时要做的内容。

在270行，可以看到一个符号P＄，我们称之为“字符串变量”。以前我们已多次使用变量，这些变量里装的是数，我们曾用盒子来作比方。字符串变量与数值变量不一样之处在于这些盒子里装的不是数，而是一些“符号”。例如下面的简单程序：

```
10  A$="GOOD  MORNING"
20  PRINT  A$
30  END
RUN <CR>
```

```
GOOD  MORNING
OK
```

可以看出上述小程序内的字符串变量 A＄内装的是 GOOD MORNING 这一串字符。我们是通过 10 行赋值语句给 A＄送“值”的。注意，字符串的值要用引号引出。

第 270 行是用 INPUT 语句从键盘输入字符串，送给字符串变量 P＄，280 行判别 P＄＝″Y″这一条件是否满足，若满足，程序就转向第 40 行重新开始。如果用户按下 N 键（或任何别的键），则 P＄＝″Y″这一条件不满足，就跳到 290 行，印出“再见”后结束程序。

在此程序中，A 和 B 两个数组用来装乘数和被乘数。用了这两个数组，每次算题后，乘数和被乘数不丢失，以备在用户答错题时把错题打印出来，所以用数组来存放这些“历史数据”。C 数组用来存放用户的答案。因为每轮为 10 道题，所以数组的最大个数都定义为 10。R 数组则用来装答错的题号。在用户答题错误时，把错题号存放在 R 数组内。R 数组也定义为最大 10，因为在最坏情况下，用户可能每一道题都答错。但如果用户答对了一部分题，则 R 数组只用到其中一部分，把数组定义大一点。不会产生语法错误的。

160、170 两行是用户答对题时的处理程序，120－140 三行是用户答错题时的处理程序，这两部分内容对某一道题来说，只能是其中一种情况，属于哪一种情况，则由 110 行来决定。如满足 C（M）＝X，则作答对处理，否则作错题处理。130 行中 M 为循环变量，M 每变化一次说明题目多了一个，所以 M 可认为是电脑出题的序号，把序号数装入 R 数组。因为 L 初值是 1（40 行），所以第一个错题序号装在 R（1）里，装完后把 L 增加

1，变成 2，下次再错一个题时，第二个错题的 M 号被装入 R（2）里了，直到循环结束。此时可知总共错了 L－1 道题。

200 行“＜ ＞”符号即数字中的“≠”符，这是 BASIC 中对≠符的表示方法。BASIC 中称为关系比较符。如果 L＝1，说明全部题目答对，就不打印错题信息了。所以这里要事先判别一下。

BASIC 中还有一些关系表示符，如表 3－2 所示。

表 3－2　　　　　　　　关系表示符

数学表示	例	BASIC	表示
＝	A＝B＋C	＝	A＝B＋C
≠	A≠B＋C	＜ ＞	A＜ ＞B＋C
＞	A＞B＋C	＞	A＞B＋C
＜	A＜B＋C	＜	A＜B＋C
≥	A≥B＋C	＞＝	A＞＝B＋C
≤	A≤B＋C	＜＝	A＜＝B＋C

用关系比较符连起来的式子叫条件表达式。如 L＜ ＞1，S＞100 等等。

第 60 行 A（M）＝RND（10）的含义是，随意产生一个 1～10 的数，把它送给组数 A 的第 M 个元素 A（M）。RND(X)是 BASIC 提供给用户的一个语句，大家可把它理解成一个“函数”来使用。括号中 X 的作用是指定上限，RND（10）即产生 0～10 的任意数，RND（100）即产生一个 0～100 的任意数。“任意”的含义是电脑每次执行这个语句时，产生的值一般都不会相等的，但是都在 0～X 的范围内。所以虽然 70 行中又有一个 RND（10）值送到 B（M），这与 60 句中的 RND（10）值一般也不会相同（不排除相同的可能性），于是 60 和 70 两行产生

了两个任意的乘数和被乘数。

这个程序用了好几个 GOTO，搞清程序执行的流转情况是十分要紧的，也是一个编程序的基本手段。

为了进一步熟悉某些语句的用法，这里提几点思考。在机器上实际试一下，马上可以见到结果。

（1）200 行改成下列写法，运行结果会怎样？

```
200  PRINT" 错误如下:"
```

（2）220 行去掉后，运行结果会怎样？

（3）如何改动程序，使“再试一轮”的用户控制达到以下目的：按 Y 键才执行下一轮，按 N 键则退出运行；按其他键则重新发问：“再试一次？”。

通过以上练习，你会感觉到程序语言的每个小变化都会产生不同的运行效果，请你根据下列几种情况来验证程序运行即可看出：

（1）全部题答对；

（2）全部题答错；

（3）答对一部分。

作为进一步的练习，你还可将此例加以改编，使它能够“升级”。如前面 10 道题全部答对（或正确率在 80%以上），可以进入下一轮，下一轮就是 1—20 的乘法运算，再下一轮是 1—30 的乘法运算等等，使程序具有如游戏中“过关”一样的效果。如果在某一轮里，正确率低于 50%则退一个难度（退回一关），这样程序的实用性和刺激性就更强了。如果编程觉得有困难，可参考第四章例 4—2 的程序。

【例 3—5】 编英文字符练习程序。可以由用户选择速度。每次以 20 个字符为一轮，一轮结束，给出正确率，并给出是否继续的询问信息。根据用户回答，决定继续运行或退出。

程序：（F BASIC、G BASIC 通过）

10 CLS	清屏幕
20 INPUT"速度(1-3)请选择";L	由用户选择速度，用 L 表示速度
30 V=0	击键正确计数
40 FOR R=1 TO 20	每轮 20 个字符
50 A=RND(89)	产生 0—89 的任意数送 A
60 IF A<65 GOTO 50	如果 A<65，重新产生，直至大于等于 65
70 X=RND(27)	产生 0—27 的任意数送 X，控制字母落下位置
80 CLS	清屏幕
90 FOR Y=1 TO 19	控制字母垂直下落的 Y 位置
100 LOCATE X,Y:PRINT CHR$(A)	让字母出现在(X,Y)位置
110 LOCATE X，Y-1：PRINT " "	字母下落
120 FOR P=1 TO L*10	L×10 控制速度
130 IF CHR$(A)=INKEY$ THEN	判别键入字母正确否，如正确，跳出 P 和 Y 循环出下一个字符
P=L*10：V=V+1：Y=19	
140 NEXT	
150 NEXT	
160 NEXT	循环 20 次，出 20 个字符
170 CLS	
180 LOCATE 10，10：PRINT	本轮正确率

```
    " 正确率"; V * 5;" %"
190  LOCATE 5, 12: PRINT                提示继续否
    " 继续下一轮 (Y/N)?"
200  LOCATE 8, 14: INPUT  P$
210  IF  P$ =" Y" THEN GOTO 10         如需要，则继续
220  LOCATE 10, 18: PRINT" 再见!"      否则退出
230  END
```

本程序可以分成三大部分，第一部分 30 行之前，做初始准备工作，我们称为“初始化”。V 记录正确键入的次数，L 表示速度，可以由用户选择 1、2 或 3。第二部分从 40 到 160 行是一个三重循环，其循环变量分别为 R、Y 和 P。第三部分从 170—230 行处理本练习报告单和提示是否继续，等待用户选择。

在分析程序之前，我们先来了解几个以前未曾见过的语句。

CHR $（A）(100 行)，从 50 行可看出 A 是一个数值。CHR $（A）在 BASIC 中称之为“ASCII（读作“阿司克”）转字符指令”。

从前面例子中我们可以看出，我们经常要与键盘打交道。从键盘来的可能是数值，也可能是符号。程序必须根据键盘的输入情况判别用户的意向以确定向何处转向。我们现在的目的是产生一个 A—Z 之间的任意字母，把它与键入的字母相比较。如两者一致就表示键入正确，否则就不正确。

我们给键盘上的字母、数字、常用符号等都编一个号码，这种号码必须全世界通用才好，否则各个国家的程序就无法通用。ASCII 码就是一种国际通用的码。ASCII 对照表可参看附录，比如大写 A 为 65，小写 a 为 97，＋号为 43 等等。

CHR $（X）的作用就是按码取字符，如 CHR $（97）就取出 a，CHR $（65）就取出 A，CHR $（44）就取出“＞”等

等。我们可以把它当成一个函数来使用。也可以进行赋值等操作。例 A$=CHR$(?)就是把 ASCII 码为?的字符送到变量 A$里。

INKEY$(130 行),在 BASIC 中称为键盘扫描命令。含义是电脑扫描键盘,看哪个键被按下,就把该键的 ASCII 码送给 INKEY$。读者会问,有了 INPUT 语句,何必还要 INKEY$?这两者一个大的差别是 INPUT 语句总是在屏幕上印出一个“?”,然后等待用户输入,并且输入完之后还要按一下回车键才继续进行下面的程序。而 INKEY$则不出“?”,也不需按回车键,INKEY$的键值也不在屏幕上显示。所以执行过程是“默默无闻”的,很适合编写本例程序。

如果在 INKEY$ 执行期间,用户没有按任何键,则 INKEY$ 装入一个空字符(ASCII 码为 32)。

回到我们的程序例上看。由于字母 A—Z 的 ASCII 码值是从 65 到 89,50 行 A=RND(89)产生一个 0—89 的随机数装入 A,而 65 以下的值不是我们练习的对象。所以在 60 行判断一下,若不落在 65—89 范围内则回到 50 行重新产生,直至符合条件。由于电脑的速度很快,所以你在运行程序时,不会感觉到第 50 行已被“白干了许多次”。

我们想让产生的字母从屏幕上方垂直落下来,而在水平方向什么位置落下来,则最好不要固定,否则过于单调,也难以显示出击键者的反应速度。所以用 X 变量来控制水平方向位置,第 70 行即产生一个 0—27 的任意数,控制起落位置。而垂直坐标应该是一个不断变化的值,才能使字符不断下落,我们用 Y 代表。根据屏幕划分,在最上方时从 Y=1 开始变化,当 Y=19 时,该字母落到屏幕下方,此时应出下一个字符了。90—150 行循环部分是控制某个字符的下落的。从 100 行可以看

出，水平位置 X 确定后，Y 每增加 1，字母就下落 1 个位置，同时我们必须“消灭”上一位置的字母，否则，随着字母的下落，将在这一个位置上充满了同样的字符，很不好看。第 110 行的任务是，在水平 X 位置的上一个 Y 值（即 Y－1）打一个空格，这样就把上一个位置的字符盖掉了。

从 120－140 行的循环，判别所键入字符是否正好是电脑产生的字符，并作相应处理。电脑产生的字符为 CHR＄（A），用户键入的字符为 INKEY＄，如果两者相等，我们要作的处理是让屏幕上的这个字符不要再落下来，而立即消失，并换一个字符从屏幕上方开始落下。就是说要跳出 140、150 行判别，两者相等后我们所作的处理是（1）P＝L×10，跳出 140 行循环；（2）Y＝19，跳出 150 行循环；（3）V＝V＋1，V 是正确击键计数器，因为又击对了一键，所以要加 1。如果 CHR＄（R）不等于 INKEY＄，说明用户击键与屏幕显示字符不一样，那么，130 行 THEN 后面部分就未被执行。循环一直进行下去，直到达到循环变量的上限，自动结束循环，再产生下一个字符。

120 行 P 循环初值为 1，终值为 L×10。我们以此来控制电脑发字符的速度，也就是练习速度。如果不是为了这个目的，就不必使用这一循环，只要保留 130 行。而去掉 120、140 行就可。因为 L 是用户选择的速度，我们把 L×10 作为“拖延时间”的指标，让 130 行的判别执行 10×L 次。如果用户是个生手，在字符下落过程中击键很慢，从前面我们已知，用户未按任何键时，INKEY＄代表空字符，所以 130 行中 IF 的条件不可能满足，这样循环就不断进行，等待用户击键。从这里我们也可以看出一个“破绽”，即如果用户选择速度时不按提示 1－3 进行，而输入一个别的值，如 5，该程序照样执行。因为 L 值越大，练习速度越慢，所以只是降低了练习速度而已。当然，也可以设

法加一些判断语句把这一破绽弥补，这一点留作读者练习。

第 3 部分从 170—230 行打出正确率报告和提供继续练习与否提示。用到了 LOCATE 语句。

LOCATE 语句用来控制背景面（见 3. 3 节）光标的定位。我们在 F BASIC 一节会看到，BASIC 语句被显示在“背景面”上，它与做游戏时的卡通图案不在同一显示面上。背景面的坐标按图 3—1 划分，X 方向从 0—27，Y 方向从 0—23，原点（0，0）在屏幕左上角。

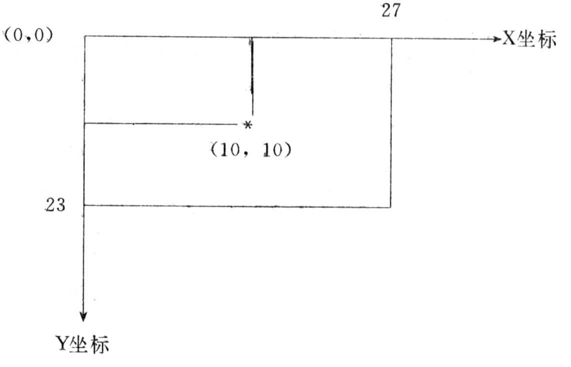


图 3—1　背景面的屏幕坐标

LOCATE 10，10 可把光标定在 X=10，Y=10 的位置上，然后 PRINT 语句在定位的光标处打印指定内容。把光标定在适合的坐标上，再打印或输入，为的是使显示看起来更醒目一些。

程序运行情况如下：

<u>RUN<CR></u>

速度（1—3）请 选择：

？3<CR>

按下回车键后，从屏幕上方开始落下字母，在出现20个字母后，如果你的击键正确率为80％（正确数为16）。显示：

正确率＝80％
继续下一轮？
？■

如按：

？Y<CR>

速度（1—3）请选择：

？■

如按N（或别的任何键），则屏幕显示再见后退出运行。显示OK。

读者可以在本例的基础上作更多的练习，使本例程序更加完善和实用，此处提供几个练习内容，建议读者自己动手试试，若有困难可参看第四章例4—3。

（1）增加一个发声提示，如果用户击键正确，就发出一个悦耳的声音，如果击键不正确，发出另一种声音。

（2）给出比本例更完善的每轮报告书，除了报告本轮正确率外，还把击错的键名报告出来，提示用户对这几个键的指法多加练习。

3.2.3 家政与管理

随着电脑走向家庭，电脑参与家政管理已经成为现实，国外和国内许多软件开发者已经开发出许多实用的软件，如“家庭医生”、“菜谱”、“健康顾问”、“家庭档案管理”等。可惜由于电脑学习机的硬件配置档次较低，特别是软件的交流方式一般限于录音磁带，而磁带上录制的程序工作速度较慢、没有自动搜索功能、内容容易丢失，大大限制了软件的开发、交流和应用。但即使如此，我们仍可利用家用电脑学习机写一些简易的程序，作为练习和应用。

【例 3—6】 计算人体节律生物钟

生理学家指出，人体生物钟有三个节律，即健康节律、智力节律和情绪节律。自出生日开始，生物钟就按照三个节律“走时”，它们的周期分别是：健康 23 天，智力 33 天，情绪 28 天。三节律分别按各自周期以正弦曲线规律变化。例如，一个出生于 1970 年 7 月 1 日的人，他的健康状况在 1970 年 7 月 7 日左右处于最佳状况，从 1 日到 7 日处于上升阶段，而在 7 日至 19 日处于下降状态，7 月 19 日左右达到最差状况，而经过一个周期（23 天）又回到一般状况，变化如图 3—2 所示。

其它两个节律也分别以各自的周期按正弦规律变化。据说有些国家已经规定，司机在情绪节律最差的几天内不得开车，以避免车祸。如果我们能查出自己某一日的健康、智力、情绪分别处于什么状况，则可有意识地加以利用。

此程序的基本要点是让用户输入自己的出生年、月、日和待查年、月、日，计算这两个日子之间已经过了多少个周期，则余下的不足整周期的天数就可很容易地推算出处于正弦曲线的

哪个位置。

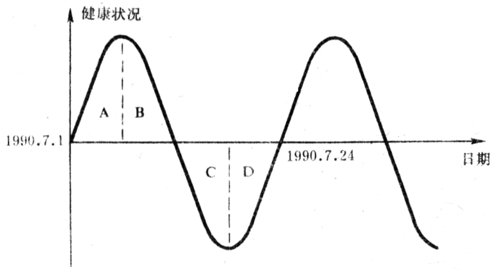


图 3—2　健康节律变化曲线

程序设计

(1) M（出生日至待测日总天数）=K（周岁数）×365+E（闰年数）+X（日数）

(2)我们把一个周期的变化分为4部分，如图3—2中的A、B、C和D所示。在A区，情况处于较好且上升阶段，在B区处于较好且下降阶段，在C区处于较差且下降阶段，在D区处于较差且上升阶段。待用户输入出生年、月、日和待查年、月、日后，由电脑告知用户该待查日处于A、B、C、D中的哪一个区。

(3) 待测年份定于1900～2000年，已知1900年为闰年。

程序：

```
10  DIM  A（3），B$（3）
20  A（1）=23：A（2）=33：A（3）=28
30  B$（1）="健康"：B$（2）="智力"：B$（3）="情
    绪"
```

```
40  PRINT " 请输入出生年，月，日"
50  INPUT N1，Y1，R1
60  PRINT "请输入待查年，月，日"
70  INPUT N2，Y2，R2
80  N=N2-N1 : Y=Y2-Y1 : R=R2-R1
90  P=0
100  FOR S=1900 TO 2000 STEP 4
110  IF S> =N1 AND S< =N2 THEN P=P+1
120  NEXT
130  M=365*N+P+30*Y+Y/2+R
140  FOR I=1 TO 3
150  Q=M-M/A (I) *A (I)
160  IF Q<=M/4 THEN PRINT B$ (I);"A" : GOTO 200
170  IF Q<=M/2 THEN PRINT B$ (I);"B" : GOTO 200
180  IF Q<=M*3/4 THEN PRINT B$ (I);"C" : GOTO 200
190  PRINT B$ (I);"D"
200  NEXT
210  END
```

对于每一次日期输入，程序要分别对三个节律计算三次，为了便于用循环语句将三次计算合成一段程序，我们采用定义两个数组的方法。数组 A 存放三个节律的周期，B$ 在 BASIC 中称为"字符串数组"。与变量一样，数组除了可以存放数值，也可以用来存放字符串，所不同之处在于要在数组的名称后面加一个"$"字符，通知电脑，该数组存放的内容是字符而不是数值。关于数组大小的定义方法则与数值数组一样，在 DIM 语句中，在数组名后加括号，括号内用数字标明数组的最多元素个数。

30 行给 B$ 的各个元素送值，其值是字符串。同字符串变量一样，值的内容要用引号引起来。因为使用的是中西文 BA-

SIC，所以字符串的内容也可以是汉字。如果使用西文 BASIC 则只能送 ASCII 字符，如字母和数字等。

本程序中，其他变量的含义如下：

N1：出生年份　N2：待查年份　N：两年份之差

Y1：出生月份　Y2：待查月份　Y：两月份之差

R1：出生日　　R2：待查日　　R：两日之差

P：　　从出生年到待查年之间的闰年数

M：　　从出生日到待查日之间的总天数

Q：　　总天数除去整周期后的余数

从 100 至 120 行之间的循环语句部分，计算出生年到待查年之间的闰年数，用 P 表示。在例 3—3 中已有了一次经验，但这里问题要复杂一些。因为在例 3—3 中计算的起点是固定的，即公元元年（闰年）。每隔 4 年，闰年数加 1 即可。而现在，我们的起点是用户任意输入的一个年份，我们不能确定这个起点年份是否为闰年。本程序采用的方法是用一把"尺子"来"量"一下，尺子的长度是四年，从 1900 年（闰年）开始量。凡是尺子头部（右端）落在出生年和待查年之外的均不算，只要尺子头部落在两者之间，则每量一次，闰年数加 1，可用图 3—3 来表示。

尺子在 A 位置时，P 不加 1，每 4 年量一次（向右移），直到尺子在 B 位置时，

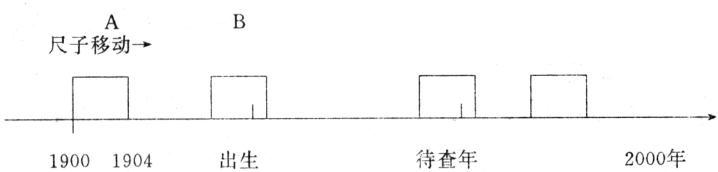


图 3—3　计算闰年数

P才开始计数。等尺子的右端移出待查年时就停止计数，这样P就把出生年和待查年之间的闰年数计算出来了。

100行以S作为循环变量，起点是1900（年），每次循环增加4（每4年有一个闰年）。然后由90行判断S的大小是否落在出生年和待查年之间，如果IF的条件满足，说明“是”，就把P增加1，然后S增加4后再判断，一直判到本程序规定的上限年份2000年为止。

110行中条件为S＞＝N1 AND S＜＝N2，这种联合条件表达式我们前面没有碰到过。“＞＝”即数学中的“≥”（大于等于），“＜＝”即数学中的“≤”（小于等于）。这里实际上有两个条件，即S必须同时满足大于等于N1和小于等于N2两个条件。对于多个条件同时满足“大条件”才满足的情况，我们可以用AND来把各个子条件表达式连起来，所以AND的含义是：AND之前和之后的条件同时满足，则该条件满足，两者之中有一个不满足，或者两条件均不满足，该条件不满足。如果我们把S＞＝N1称为条件A，S＜＝N2称为条件B，S＞＝N1 AND S＜＝N2称为条件C，则三个条件的关系可用表3－3表示。若条件成立，我们称为真，否则为假。

表 3－3　　　　三个条件的关系

条件A	条件B	条件C
真	真	真
真	假	假
假	真	假
假	假	假

从表3—3可以看出，只有当条件A和B都成立时，条件C才成立。这就是用AND连接条件式时的关系，我们称这种关系

为“逻辑关系”，用 AND 的逻辑称为“与逻辑”。

计算出闰年数后，再算天数就容易了。第 130 行 M 即代表出生日到待查日之间的天数，由下列几项组成。

（1）365×N　N 年乘以每年 365 天；

（2）P　闰年数，每个闰年多一天，共 P 天；

（3）30×Y　每月按 30 天计算，差 Y 月。Y 可能是正的也可能是负的，例如一个出生于 1970 年 8 月 1 日的人，要查 1972 年 5 月 23 日状况，按上述计算方法 N＝1972－1970＝2，在第（1）项中计算了两个整年，实际上只有 1 年多时间，所以此处 Y＝5－8＝－3，扣除 3 个月的天数；

（4）Y/2　考虑到有 31 天的月份，所以补上 Y/2 天，这部分不严格，未考虑 2 月份及上半年、下半年的差别；

（5）R　所差天数。同（3）一样，R 也可能是正的或负的。

从 140 行至 200 行之间的循环共进行了三次，第一次时 I＝1，A（1）＝23，所以计算的是健康节律；当 I＝2 时，A（2）＝33，计算的为智力节律；I＝3 时显然计算情绪节律。从这里我们也可以看出使用数组来装数的好处，可以用一个变动的标号来调用这些数。如果我们用三个不同的变量而不使用数组，计算 Q 的 150 行就要写三遍才能分别计算出三个不同的节律。第 150 行是针对整数 BASIC 的特点写的。如果是浮点 BASIC 就不能这样写，因为对于浮点 BASIC 来说 M 除以 A（I）又乘以 A（I）等于 M 本身。对于两种 BASIC 来说，全部程序仅此一句需要改写，别处均可通用。

运行情况如下：

```
RUN <CR>
```

出生年，月，日？

？1970，8，1<CR>

待查年，月，日？

？1972，5，23〈CR〉

体力 A

智力 A

情绪 A

从上述分析我们也可看出，这个程序有几个不足之处。

（1）用 4 个区来指示节律情况比较粗糙，而且难以看出待查日前后几日的情况；

（2）待查年上限限制于 2000 年以内，应用范围较狭，是否可延展到任意年。

读者不妨针对上述不足，自己设法改进，使程序的应用更为方便。若有困难，可参考第四章例 4－3。

【例 3－7】 用儿童足长预测成年后身高。

孩子将来的身高能长多高，这是每个家长关心的问题。中国青少年体质研究组根据对十一万余名城市七岁至二十五岁男女学生的测试数据，研究出了根据儿童的足长预测成年后身高的方法，并指定了足长预测身高表（见本书附录）。

足长的测量为足后缘突出处至最长趾端的垂直距离。可用有毫米刻度的直尺测量。将直尺垂直平放于地面，一端靠抵墙根，受测者右脚踩于尺上，足跟后缘抵墙，足的纵轴与直尺平行，然后用一挡板与直尺垂直并抵最长距离读数。

该表分男女二组，可根据儿童的性别分别查阅。垂直方向找到儿童足长，水平方向找到儿童年龄，两条直线交叉处即为儿童成年后身高。

例如预测一个七岁，足长 192 毫米的男孩身高，可查男子

足长预测身高表。在足长栏内找到 192 毫米栏，在年龄栏内找到 7 岁栏，两栏交叉处读数为“177”，即该儿童成年后身高为一百七十七厘米。

以上资料选自叶恭绍、朱智贤、林巧稚等主编的《家庭育儿百科全书》，北京出版社 1990 年 5 月出版。

如果我们把这张足长预测身高表编制成程序，就可随意向电脑查询，而不必每次去翻书查表，并且可让儿童自己测量足长，自己通过电脑查询，通过这一过程，培养对电脑的认识和兴趣。

这一程序的目的还在于让读者认识怎样把表格录入到计算机程序内，程序怎样“查表”。因为程序查表是经常要做的工作。例如，在统计分数时，查出 90 分以上的同学等等。所以本程序的编制虽然不难。读者却应注意这一方法。

为了说明问题方便起见，本程序只编入 7、8、9 三年龄的表格资料，读者如有兴趣，可按照这一方法将它扩充到全部年龄。

程序分析与设计：

（1）我们可以设计用 6 个数组，分别来装 7 岁、8 岁、9 岁的资料。由于分男、女两个表，所以一共需要 6 个数组。分别定义为 A1、A2、A3 和 B1、B2、B3。

（2）每个数组数据的个数不一样，例如男 7 岁组共计 27 个数据，男 8 岁组共计 31 个数据等。数据最多的为男 9 岁组，为 36 个数据。

（3）可以用 READ－DATA 语句将表中数据列于 DATA 语句中，在程序内用 READ 语句读入到相应数组中去。

（4）查找数据元素的序号。该序号必需与足长挂钩，例如前面提到的 7 岁男孩，足长 192 毫米，从表上看，该数据在 A1

数组的第 12 个元素内，如何从 192 毫米产生出下标号 12 呢?

我们观察足长的变化是十分有规律的，即每隔 2 毫米为一栏，起点处为 170，所以只要计算待测者的足长与 170 之间有多少个 2 就可算出它的序号。正因为如此，也不必将所有足长数据用数组来录入。

（5）为了便于在整数 BASIC 上运行，避免小数输入，足长单位采用毫米，凡输入足长的毫米数为奇数者（例如 191 毫米），由于表上无相应身高数据，本程序采用舍掉 1 毫米的方法，按 190 毫米预测，所以结果偏保守一点。读者也可采取求平均的方法，即对 191 毫米足长，查出 190 和 192 两个身高数，然后将两个数加起来除以 2，按平均身高来预测。

程序：

```
10  DIM A1（27），A2（31），A3（36）
20  DIM B1（26），B2（31），B3（35）
30  Z$ ="成年后身高预测为"
50  FOR I=1 TO 27 ：READ A1（I）：NEXT                读数据
60  FOR I=1 TO 31 ：READ A2（I）：NEXT
70  FOR I=1 TO 36 ：READ A3（I）：NEXT
80  FOR I=1 TO 26 ：READ B1（I）：NEXT
90  FOR I=1 TO 31 ：READ B2（I）：NEXT
100  FOR I=1 TO 35 ：READ B3（I）：NEXT
120  INPUT "请输入性别(男～1,女～2)";A        A 为性别代码
130  IF A< >1 AND A< >2 THEN GOTO 120
140  INPUT " 请输入足长（毫米)"；B                B 为足长
150  INPUT "请输入年龄(7—9)";X                  X 为年龄
155  IF X<7 OR X>9 THEN GOTO 150
160  R=（B—170）/2+1                         R 为计算出的序号
```

```
170 IF A=2 THEN R=R-1 : GOTO 210   女孩从 210 行开始
180 IF X=7 THEN PRINT Z$ ; A1 (R): GOTO 240
190 IF X=8 THEN PRINT Z$ ; A2 (R): GOTO 240
200 PRINT Z$ ; C (R) : GOTO 240
210 IF X=7 THEN PRINT Z$ ; B1 (R): GOTO 240
220 IF X=8 THEN PRINT Z$ ; B2 (R): GOTO 240
230 PRINT Z$ ; B3 (R)
240 END
250 DATA 157, 159, 160, 162, 164, 166, 168, 170, 172, 173,
    175, 177
260 DATA 179, 181, 183, 184, 186, 188, 190, 192, 194, 195,
    197, 199
270 DATA 201, 203, 205
280 DATA 152, 154, 155, 157, 159, 161, 163, 164, 166, 168,
    170, 171
290 DATA 173, 175, 177, 179, 180, 182, 184, 186, 188, 189,
    191, 193
300 DATA 195, 196, 198, 200, 202, 204, 205
310 DATA 146, 147, 149, 151, 153, 154, 156, 158, 159, 161,
    163, 165
320 DATA 166, 168, 170, 171, 173, 175, 177, 178, 180, 182,
    183, 185
330 DATA 187, 189, 190, 192, 194, 195, 197, 199, 201, 202,
    204, 206
340 DATA 151, 153, 155, 156, 158, 160, 162, 163, 165, 167,
    169, 170
350 DATA 172, 174, 176, 177, 179, 181, 183, 184, 186, 188,
    190, 191
360 DATA 193, 195
370 DATA 145, 147, 149, 150, 152, 154, 156, 157, 159, 161,
```

```
     162, 164
380  DATA 166, 167, 169, 171, 172, 174, 176, 178, 179, 181,
     183, 184
390  DATA 186, 188, 189, 191, 193, 194, 196
400  DATA 140, 141, 153, 144, 146, 148, 149, 151, 152, 154,
     156, 157
410  DATA 159, 161, 162, 164, 165, 167, 169, 170, 172, 174,
     175, 177
420  DATA 178, 180, 182, 183, 185, 187, 188, 190, 191, 193,
     195
```

以上程序结构并不复杂，也很容易读懂。有关DATA语句的使用这里略加说明：

（1）DATA语句中可放入多个数据，但是每行最后一个逗号不可以加，READ语句按照读数的顺序，先后从DATA语句中去取数。电脑内部有个“指针”会自动记忆取到哪一个了，下次再碰到READ时，从指针的下一个数开始读，所以第50行的循环读走了DATA语句前面27个数，指针指向数据27处，第60行的READ语句则从28个DATA数据开始，又从280行依次取走31个数据，直到全部READ语句执行完毕。

（2）DATA语句中数据个数不得少于READ语句执行次数，否则程序出错。

（3）DATA语句可以放在程序的任何位置，本例中放在240行END之后，也不影响执行。

关于程序的执行也有几点请注意。

（1）155行是用来控制用户对年龄的选择，低于7或高于9时，程序将回到150行，提示用户重新选择。只有用户按下7—9之间的数，程序才可能往下进行。

前面例 3—6 中，把两个条件用 AND 连起来组成一个总的条件，称为“与逻辑”。这里我们把两个条件即 X＜7，X＞9 用 OR 连起来，则称为“或逻辑”。如果我们也像例 3—6 一样，用条件成立为“真”，不成立为“假”，并把两个子条件分别用条件 A 和条件 B 来表示，那么，总条件 C 与条件 A、B 的关系如下表 3—4 所示：

表 3—4　　　　或逻辑

条件 A	条件 B	条件 C
假	假	假
假	真	真
真	假	真
真	真	真

从这个表可以看出，X＜7、X＞9 中只要有 1 个成立或两者同时成立，OR 组成的条件成立，就要转向 150 行重新发问。显然这里如果用 AND 连接，则会发生逻辑错误，因为不可能找到一个数，同时满足小于 7 和大于 9 两个条件，于是总条件永远得不到满足，那么，不论对 X 作什么输入都不会执行 GOTO 150 语句，而是直接转向 160 语句去执行了。语句 130 中的逻辑采用 AND 连接的逻辑，读者可自行分析其原因。

（2）序号 R 的计算，对男孩和女孩计算数组元素序号 R 的公式是不一样的，因为男孩足长的起点是 170mm，女孩起点是 172mm，两者的序号差 1。例如对于足长 172mm 的 7 岁孩子，女孩的序号为 1，而男孩的序号为 2。所以在 160 行计算出男孩序

号后，若170行判断为女，则将R－1后转向210去查阅女孩的表，否则直接查阅男孩的表（180～200行）。

读者读懂此程序后，可思考下列问题：

（1）为什么不在一个循环语句内将所有数据读完？

（2）能不能改造程序写法，用一个循环语句读完？

（3）本程序使用了六个数组，占用了大量内存单元，而实际每次运行查阅的仅是六个中的一个，是否有可能少用些数组？

上述问题可参考第四章例4－5。

3.3 F BASIC 编程

电脑学习机和普通电脑一个最显著的区别就是在电脑学习机上可以使用游戏卡，这是苹果机和其他类型的家用电脑都不具备的特点。为了将这一特点更加“淋漓尽致”地加以发挥，人们特地研制了一种在学习机上使用的BASIC语言，称为F BASIC（Family BASIC）即家庭BASIC。F BASIC是建立在整数BASIC基础上的，它与普通整数BASIC不同之处在于它的谱曲和游戏功能特别强。它允许用户以BASIC编程的方法在程序里谱曲，设置了特有的谱曲语句PLAY，当运行程序时，电脑便自动按程序奏乐。F BASIC也可以方便地编制游戏程序，直接用语句调用它的卡通，用游戏手柄指挥卡通行走、跳跃等。

由于上述特点，一般电脑学习机都配备有F BASIC。有的机器配备G BASIC（如小霸王学习卡1．0版，3．0版），其使用与F BASIC基本相同。

3.3.1 谱曲程序

我们在第二章已熟悉音乐板的使用，我们可以在音乐板中写上一些曲子，并让电脑演奏。但这些曲子不是以程序的形式出现的，一旦关掉电源再次开机，已经谱写的曲子将失去，即音乐板中的曲子不能存储。再有，如果我们想在某一个程序中加一段乐曲，如在例 3－4 乘法练习中加两段不同的曲子，若答对，奏一段乐曲 A，答错则演奏乐曲 B，我们同样不能依靠音乐板来实现这一目的，而用程序谱曲，则可克服以上两个困难。

若将下列程序输入学习机，并加运行，就可演奏《美国人的羊》。

```
5  REM AMERICAN SHEEP
10  CLS
20  LOCATE 8，12 ：PRINT ″ AMERICAN SHEEP″
30  PLAY ″M1Y2V7T3 ：M1Y1V5T3 ：M1T3″
40  PLAY ″O2A6G3F5G″
50  PLAY ″A5AA7″
60  PLAY ″G5GG7″
70  PLAY ″A5O3CC7″
80  PLAY ″O2A6G3F5G″
90  PLAY ″AAA7″
100  PLAY ″G5＃AA6G3″
110  PLAY ″F9″
120  END
```

从上面程序可以看出，演奏程序主要使用了 PLAY 语句，

在 F BASIC 中，利用 PLAY 指令可以演奏具有八种节拍变换，四种变色效果，16 节音量、音长变化，6 个八度音，72 键钢琴旋律的三重奏乐曲。PLAY 指令可把我们带入电脑乐曲的美妙境界。这也是 F BASIC 的一个特色。

PLAY 语句的格式为 PLAY“音乐字符”。

如上例第 30 行中引号内的内容即为音乐字符，或称“弦乐器”。所以我们要谱曲，就要搞清楚什么是音乐字符。

一、音域、音阶、音符长度

对于每一个音符，要在 PLAY 语句中指定音域、音阶和音符长度的代码。

7 个音符 1，2，3，4，5，6，7（即 DO，RE，MI，FA，SO，LA，TI）分别用字母 C、D、E、F、G、A、B 表示。

音域的代码用“OX”代表，X 从 0～5，如表 3－5 所示。

表 3－5　　　　音域代码表

音阶	1～7 · · · · · ·	1～7 · · · ·	1～7 · ·	1～7	· · 1～7	· · · · 1～7
音域代码	O0	O1	O2	O3	O4	O5
	低　音　区			中音区	高　音　区	

高音阶表示比基本音高半音，在乐理中升记号为＃，如＃5 称为“升 5”，下面列举一些 BASIC 的音阶代码例。

音阶 1　　＃1　　＃2　　4　　＃6　　0 休止

代码 C　　＃C　　＃D　　F　　＃A　　R0－R9

休止长短用 R0－R9 表示，R0 最短，R9 最长。

音阶长度在简谱中用“－”表示，代表音的长短。5 表示四分音符；$\underline{5}$ 表示八分音符；5－表示二分音符等等。BASIC 的音

符长度代码如表 3--6 所示。

表 3—6　　　　音符长度代码

代码	音符	拍节	示例
9	全音符	4	1———
8	付 2 分音符	3	1——
7	2 分音符	2	1—
6	付 4 分音符	1　1/2	1.
5	4 分音符	1	1
4	付 8 分音符	3/4	1.
3	8 分音符	1/2	1
2	付 16 分音符	3/8	1.
1	16 分音符	1/4	1
0	32 分音符	1/8	1

按照上述规定，我们可以为简谱写出其 BASIC 代码。

【例 3—8】　写出下面音符的 BASIC 代码。

（1）5.　　O3G4　　O3：中音区，G：DO 的音阶代码

4：8 分音符

（2）6　　O3A5　　A：LA 的音阶代码　　5：4 分音符

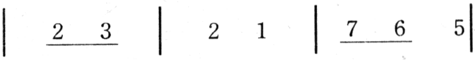
（3）5. 5　6　5　　O3G4G1A5G5

（4）17—　　O4C4O3B7

上例（3）我们用 O3 指定了所有音符，而（4）中用 O4、O3 两个音部。在 F BASIC 中规定：如果后面的音符与前面的在同一音部，则音域代码可略去不写。

例（4）中由于 i 和 7 不在同一音部，所以后面要重写，而(3) 中所有音符都在同一音部，故用 O3 指定后，后面音符不必再重复。

对于音符长度代码，后面与前面相同时也可以作同样省略。如：



可编写为

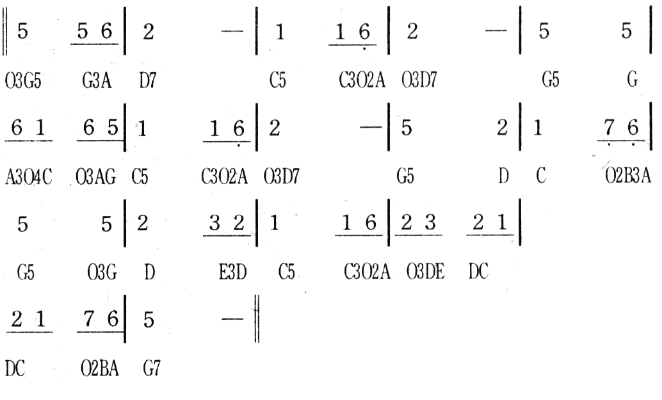
O 3 D 3 E D C O 2 B A C 4

而不必写为

O 3 D 3 E 3 D 3 C 3 O 2 B 3 A 3 C 5

【例 3—9】 编写《东方红》的乐曲程序。

简谱如下，在简谱下我们按规则填上代码。



程序如下：

```
10  PLAY "O3G5G3AD7C5C3O2AO3D7"
20  PLAY "G5GA3O4CO3AGC5C3O2AO3D7"
30  PLAY "G5DCO2B3AG5O3GDE3DC5C3O2A"
40  PLAY "O3DEDCDCO2BAG7"
50  END
```

可将程序输入电脑学习机内，运行并修改错误，然后录入录音磁带，以备再用时从磁带上调入学习机。

二、速度、音量、音色效果

速度、音量、音色都可以在程序中加以指定，使乐曲的演奏更加随心所欲。

（1）速度

用 T 设定，代码从 T1～T8。T1 最快，T8 最慢。

（2）音量

用 M 表示有无顿音。M1 为有顿音，M0 则为一般。

用 V 表示顿音效果或音量大小，组合情况如下：

M1Vn：有顿音，V1～V4 时顿音明显，V5～V15 时只有短音效果，音量固定（＝15）。

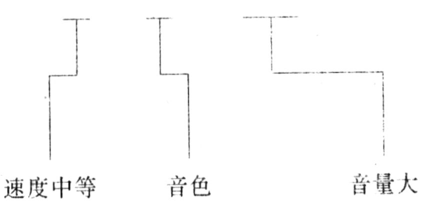
M0Vn：一般情况，Vn 可规定音量 V1（最小）～V15（最大）。

（3）音色

用 Y 表示，分为 Y0～Y4 四档音色。

如果我们在例 3—9 程序头部加一行：

```
5  PLAY "T4     Y2     M0V15"
```



则以下程序行中所有 PLAY 语句均按此行指定的值来演奏。你可改变其中的参数值。体会一下各种不同的速度、音量、音色

效果。

【例 3－10】 编写“祝你生日快乐”程序

简谱：

C 3/4 祝你生日快乐

| 5·5 6 5 | 1 7 — | 5·5 6 5 | 2 1 — |

| 5·5 5 3 | 1 7 6 | 4·4 3 1 | 2 1 — |

音符代码：

|O 3 G 4 G 1 A 5 G 5|O 4 C 5 O 3 B 7|G 4 G 1 A 5 G 5|

|O 4 D 5 C 7|O 3 G 4 G 1 O 4 G 5 E 5|C 5 O 3 B 4 A 5|

| O 4 F 4 F 1 E 5 C 5 | D 5 C 7 |

根据上面已讲过的规定，如果后面音符与前面音符长度一样，则后面的音符长度代码可省略。即上面代码中下标有下划线者均可略去不写。上面乐谱可简化为下面形式：

|O 3 G 4 G 1 O 4 G 5 E|O 4 C 5 O 3 B 7|G 4 G 1 A 4 G|

|O 4 D 5 C 7|O 3 G 4 G 1 O 4 G 5 E|C 5 O 3 B A|

| O 4 F 4 F 1 E 5 C | D 5 C 7 |

BASIC 程序如下：

```
10  PLAY "M1V10Y2T4"
20  PLAY "O3G4G1A5G"
30  PLAY "O4C5O3B7"
40  PLAY "G4G1A5G"
50  PLAY "O4D5C7"
60  PLAY "O3G4G1O4G5E"
70  PLAY "C5O3BA"
80  PLAY "O4F4F1E5C"
90  PLAY "D5C7"
100  END
```

若把 100 行 END 改成 GOTO 20，则乐曲反复演奏，直到按 Esc 键，强制退出程序运行状态。

三、缺省

在例 3—8 中，我们在程序开始时并未指定曲子的速度、音量、音色等，也可进行演奏，这是因为在 BASIC 中已经规定，如不加任何指定，曲子自动被指定为下列参数：

T4M0V15O3C5R5

一旦指定参数后，除非遇到新的指定，否则参数值维持不变。

F BASIC 提供给用户的谱曲编程功能还可以编制二部曲三部曲等重奏曲，有演奏各种和弦的功能。读者如有兴趣，可参考更详细的资料介绍。

3.3.2 游戏程序

除了谱曲，F BASIC 还有很强的游戏功能。喜欢电脑游戏的读者心目中早已有许多熟悉的“人物”：玛丽兄弟、企鹅、乌龟等等。我们称之为“卡通”，就像舞台上的演员，我们可以在程序中“调动”这些演员，表演各种动作。对于青少年读者来说，不仅具有很强的趣味性，而且在不知不觉中，对程序的编写变得精通起来。

在 F BASIC 中要完成游戏程序，比完成乐曲谱曲需要熟悉更多的语句和命令。譬如怎样配背景颜色，怎样变换卡通的姿势，怎样确定行走路线等。好在我们已经有了前面若干编程的训练，再来熟悉这些命令和语句就不会感觉困难了。

一、如何调出卡通

观众看演员演出，首先要把幕拉开，否则看不到演员。同样，要使卡通出现在屏幕上，就必须拉开卡通的幕，我们称它为“卡通面”。F BASIC 的显示面共有 4 个，就好像一个舞台上设置多层幕布一样，各显示可以分层展开。四层显示关系如图 3—4 所示。

第一层称为卡通 0 面。卡通的活动都在这一面上进行。

第二层为背景面。我们在打开卡通面后，仍可显示 BASIC 程序，并作修改和增补等。这些都在背景面上。游戏的背景如建筑物、大山等也在这一面上。背景面的坐标划分我们已有所介绍。

第三层为卡通 1 面。玩游戏的读者都有经验，有时候，卡通会走到背景后面去，但电脑不能因为它走起来就把它“忘

了”，需要时仍要让它出来活动，实际上它处于卡通 1 面。

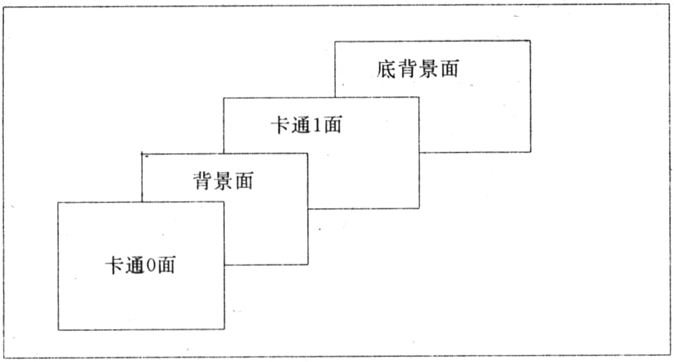


图 3—4 各屏幕显示层面

第 4 层为底背景面。适合表示天空、陆地和大海的效果。这一面初始时被设定成黑色，你可以在程序中在需要的地方改变成你想要的颜色。

本书主要叙述卡通在 0 面上的调用和活动方式，使你获得游戏程序编制的基本概念。更详细的介绍可参考有关的资料。

调出卡通要分三步：

1. SPRITE ON 语句

SPRITE ON 语句的作用是打开卡通面，SPRITE 英语原意是“小精灵”，这里是卡通的意思，ON 就是打开。SPRITE ON 也可作命令用。

2. DEF SPRITE A，(B，C，D，E，F) =CHR $ (N1) +CHR $ (N2) +CHR $ (N3) +CHR $ (N4)

幕拉开了，但是让哪个演员出场和以哪种姿势出场呢？这就必须事先进行指定。DEF SPRITE 就是指定某个卡通及其姿

势，我们称之为定义卡通命令，DEF 就是定义的意思。举例说明如下：

DEF SPRITE 0，（0，1，0，0，0）＝CHR＄（0）＋CHR＄（1）＋CHR＄（2）＋CHR＄（3）

上面这个命令 SPRITE 后面，有 6 个数字，数字之间用逗号隔开，每个数字都代表一种意思，其含义如下：

0(0， 1， 0， 0， 0)

代表卡通图案是否上下翻转
也就是说决定卡通是“头朝下”，还是“头朝上”，0代表不翻转，1则代表翻转。

同上，代表卡通图案是否左右翻转，0代表不翻转，1代表翻转。

表示卡通将出现在卡通0面还是1面。0表示0面，1则表示1面。有关卡通面的概念将稍后再解释。

代表卡通的大小，如果为0，则卡通只有一个字符大小，如果为1，则有4个字符那么大。

颜色代码，可为0、1、2、3。

卡通号，可为0、1、2、3、4、5、6、7，出场的每个卡通都有一个编号，称卡通号，这样，在有多个卡通时，只要指定一个号，就可对相应的卡通进行操作。

DEF SPRITE ON 命令后的 6 个数字，统称为它的参数。F BASIC 中很多命令后都有参数，有的只有一个，而有的则有多个。DEF SPRITE 命令就有多达 6 个参数，这 6 个参数各自代表的含义上面已作了介绍。看起来很复杂，实际并不难，只要反复练习几次就得心应手了。

上面的命令，形式上像个等式，左面的意思根据上面的介绍，解释为：0 号卡通的颜色用 0 号色，大小为 4 个字母大小，将它调用到卡通 0 面上，左右、上下均不翻转。细心的读者也许会问：0 号卡通是什么？等式左面并没有说明呀？对，等式左面只是指明卡通的颜色、大小、姿势等等，并没有说明是什么，而等式的右边则加以说明。每个卡通都有一个代码，各个卡通的代码可参见附录 5。将代码按顺序填入上述命令右面的括号内，就指定了卡通。上述命令中填入的是 0、1、2、3，就指定了 0 号卡通是玛丽沃（走 1）。

3. SPRITE N，X，Y

幕拉开了，演员及其姿势也确定了，下一步就该指定演员在舞台上的位置了。这可以用 SPRITE N，X，Y 来完成，其中 N 为卡通号，X 为水平方向位置，Y 为垂直方向位置。举例说明如下：

```
        (N)   (X)    (Y)
SPRITE  0,    140,   120
```

- 120：卡通位置的Y坐标为120
- 140：卡通位置的X坐标为140
- 0：卡通号为0

图 3—5 表示出卡通面屏幕画面的坐标，注意与图 3—1 背景面坐标的区别。

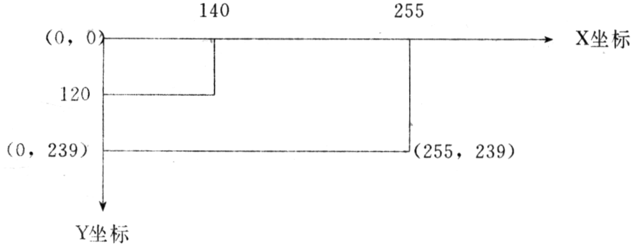


图 3—5 卡通面坐标

将屏幕的横向位置用 X 坐标表示，X 最小为 0，最大为 255，若超出该范围，电脑将给出错误信息。纵向位置用 Y 表示，Y 最小为 0，最大为 239。改变 SPRITE N，X，Y 中的 X 和 Y，可改变卡通的位置。

4. 调出卡通的程序

【例 3—11】 调出玛丽沃。

有了以上三步就可以调出卡通了，先用 NEW 命令清除以前的程序，然后输入以下程序调出玛丽沃：

```
10  CLS                               清除屏幕，使画面显得干净
20  SPRITE ON                         打开卡通面
30  DEF SPRITE  0,(0,1,0,0,0)  定义 0 号卡通为玛丽沃（走 1）
=CHR$ (0) +CHR$ (1) +CHR$ (2) +CHR$ (3)
40  SPRITE 0, 140, 120                确定卡通的位置，并让卡通显示出来
```

现在请运行一下，看是否出现卡通图案：

RUN<CR>

OK

如果你输入程序无错误，在屏幕的中央出现玛丽沃（走 1）的图案。

输入第 30 行时，将超过屏幕一行的长度，输入时不必理会，直到全屏输完，最后再按 Enter 键。

在输入程序时，为提高输入速度，F BASIC 中有些字符串可以用一个键代替。如："SPRITE ON" 可用功能键 F5 代替，至于其它功能键（F1～F8）的代替如下：

F1	MOVE	F5	SPRITE
F2	PRINT	F6	CONT
F3	GOTO	F7	LIST
F4	CHR $ (	F8	RUN

如果你忘了他们之间的对应关系，你只要按：

KEY LIST

就可在屏幕上查看到。所以记住 KEY LIST 命令是非常有用的。

现在让我们来使用一下清屏幕命令：

CLS<CR>
OK

你可以发现，屏幕上的卡通并没有消失，要使卡通消失可用 SPRITE OFF 命令，请试一下：

SPRITE OFF<CR>
OK

现在卡通在屏幕上消失了。

该命令与 SPRITE ON 是一对,OFF 的意思是关闭,ON 的意思是打开。

现在再输入命令:

SPRITE ON<CR>
OK

你发现卡通又出现了，说明 SPTITE OFF 并没有真正将 0 号卡通从电脑中抹去，而只是不让它显示出来而已，若要真正将其抹掉，需用下列命令:

SPRITE 0<CR>
OK

该命令后的数字，就是你想抹去的卡通号，这里是 0，所以抹掉了 0 号卡通。若再用 SPTITE ON 命令，将看不到 0 号卡通。你可用 SPRITE N 抹掉 N 号卡通。

SPTITE N（N 可以是 0、1、2・・・7）与 SPTITE OFF 的区别有点像 NEW 与 CLS 的区别。NEW 能清除电脑记忆的程序，而 CLS 只能清除屏幕。同样道理，SPTITE N 可以清除电脑记忆的卡通，而 SPRITE OFF 只能关闭卡通显示。

对程序稍加修改，就可调出不同的卡通，也可以改变卡通的位置。例如，将第 30 句右面括号中的四个数字依次改为 104、105、106、107，运行一下，就会发现玛丽沃变成了企鹅（正

面），再将第 40 句中的 140，120 改为 100，200，运行后发现企鹅的位置变了。请读者多试几次，对程序进行改动，从而加深印象。

二、让卡通活动

1. 改变卡通位置的方法

【例 3－12】 让卡通在随意位置出现。

程序 A：

```
10  CLS
20  SPRITE ON
30  DEF SPRITE 0,(0,1,0,0,0)=CHR$(0)+CHR$(1)
    +CHR$(2)+CHR$(3)
40  X=50
50  Y=120
60  SPRITE 0, X, Y
```

该程序的作用与程序例 3—11 的作用完全一样，这里的第 40、50、60 句相当于程序 3—11 中的第 40 句。现在修改第 40、50 句，使 X＝200，Y＝200，运行一下，发现卡通的位置改变了。这说明用变量表示位置，只要改变变量的值就可以改变卡通的位置。该程序中，给 X、Y 送数，采用的是赋值语句，想改变变量必须修改程序，这很不方便，INPUT 是从键盘送值的命令，在这里使用就比赋值语句方便。

现在请将程序的第 40 和 50 句改为：

```
40  INPUT " X="; X
50  INPUT " Y="; Y
```

然后运行一下：

RUN<CR>

则出现：

X=?
100<CR>
Y=?
200<CR>

即当你给定坐标（100，200）后，卡通移到相应的位置，再运行一下：

RUN<CR>
X=?
50<CR>
Y=?
0<CR>

这时卡通又跑到屏幕的上部。

上述方法要求每次改变都须运行一下程序。能否连续运行呢？请在程序中加入一句：

```
70 GOTO 40
```

加入该语句后程序成为程序 B：

```
10 CLS
20 SPRITE ON
30 DEF SPRITE 0, (0, 1, 0, 0, 0) =CHR$ (0) +CHR$ (1)
  +CHR$ (2) +CHR$ (3)
40 INPUT " X="; X
50 INPUT " Y="; Y
60 SPRITE 0, X, Y
70 GOTO 40
```

加上第 70 句后，程序将不会结束，我们称为“死循环”，要结束程序。必须按“Break”键，这时屏幕出现“BREAK IN 40”，告诉你程序是在第 40 句中断运行的。

2. 让卡通自己运动

在例 3－12 中，操作者每输入一个坐标，卡通的位置就跟着改变一次。下面的程序可以使卡通自己运动起来。

【例 3－13】 卡通自己活动

```
10 CLS
20 SPRITE ON
30 DEF SPRITE 0, (0, 1, 0, 0, 0) =CHR$ (0) +CHR$ (1)
  +CHR$ (2) +CHR$ (3)
40 X=200
50 FOR Y=0 TO 200
60 SPRITE 0, X, Y
70 NEXT
```

程序中，第 50～70 句中的 FOR-NEXT 循环，循环部分要重复 201 次，第一次 Y=0，第二次 Y=1，第三次 Y=2…… ……，第 201 次 Y=200，每次执行到第 70 句的 NEXT，变量 Y 的值

就增加1，并返回第50句继续执行，直到Y大于200。执行一下该程序：

RUN<CR>

你将会发现，卡通玛丽沃快速地从屏幕上方垂直降落在下方，这是因为第60句中的SPRITE 0，X，Y要重复201次，其中X的值一直为200，即卡通的横坐标不变，而Y每次加1。即该句执行的是SPRITE 0，200，0、SPRITE 0，200，1、SPRITE 0，200，2.……SPRITE 0，200，200，因此，玛丽沃的位置实际上改变了200次。

如果想让玛丽沃运动得慢一点，请加入一句：

```
65 PAUSE
```

这里PAUSE的意思是暂停。程序每次执行到这一句时会暂停下来，操作者只要任意按下一个键，比如空格键，程序就会继续执行，请运行一下试试。

这时发现玛丽沃停在屏幕上方不动，按一下空格键（或其它任意键），玛丽沃向下移动了一点，若连续按下空格键，玛丽沃将可连续向下运动。这里PAUSE后面没有参数。如果有参数，例如PAUSE 5，则表示暂停5个单位时间后自动继续，请将第65句改为：

```
65 PAUSE 5
```

运行一下，将发现玛丽沃以较慢的速度自上而下运动，若

改为：

```
65 PAUSE 10
```

运行后，可以看到，玛丽沃运动得更慢。

3. 用操纵器控制卡通运动，STICK 语句

例 3—13 程序可以使卡通自己运动，现在我们来学习如何使用操纵器控制卡通运动。首先介绍一下操纵器指令。

我们知道游戏机都有两个操纵器，又叫操作手柄，分为 1 号和 2 号，上面有十字形的方向键，玩游戏时，通过按动方向键，可以使卡通朝不同的方向运动，电脑是如何知道操作者按动了十字方向键的哪一侧呢？那就是通过 STICK 命令。它有两种形式：STICK（0）用来检测 1 号操纵器的方向键盘是否按下，而 STICK（1）用来检测 2 号操纵器的方向键盘是否按下。

为了说明方便，不妨只使用 1 号操纵器，令 S＝STICK（0），则 S 的值按下面确定：

如果方向键未按下则 S＝0；

如果按下方向键的右侧，则 S＝1；

如果按下方向键的左侧，则 S＝2；

如果按下方向键的下侧，则 S＝4；

如果按下方向键的上侧，则 S＝8。

我们可以使用 IF － THEN 语句来控制卡通的运动，例如：

```
10 IF S=1 THEN X=1
20 PRINT X
```

从上面这两行程序我们可以知道：（1）如果 S＝1 成立，则

让 X=1，然后执行第 20 句的 PRINT X。(2) 如果 S=1 不成立，也就是说 S≠1，则不执行 X=1，直接执行第 20 句的 PRINT X。

有了这个命令，我们可以用 IF — THEN 语句编写一段程序，用操纵器控制卡通的运动。

【例 3—14】 编写一段程序，可以用操纵器控制卡通运动：

10 CLS	清屏幕
20 SPRITE ON	打开卡通面
30 DEF SPRITE 0,(0,1,0,0,0) =CHR $(0)+CHR $(1)+CHR $(2)+CHR $(3)	定义 0 号卡通为玛丽沃
40 X=120	让 X 坐标的初始值为 120，即从屏幕的中部开始运动
50 S=STICK (0)	测试 1 号操纵器的方向键是否按下，结果放在变量 S 中
60 IF S=1 THEN X=X+1	如果按下方向键的右侧，即 S=1 成立，那么让 X 坐标位加 1，即让卡通向右走一格，否则继续执行 X 值不变
70 IF S=2 THEN X=X-1	若按下方向键的左侧，即 S=2 成立，那么让 X 坐标值减 1，即让卡通向左走一格
80 IF X>250 THEN X=10	如果 X 大于 250，那么让 X=10，随着 X 的值不断增加，即不断地按向右键，则卡通将跑到屏幕的最右边，直至看不见。甚至当 X>255 时将超出 SPRITE 0，X，Y 的范围，从而导致错误，

	所以加以限制，即当玛丽沃跑到最右边时，让它再从左边出现
90 IF X<10 THEN X=250	道理同上，如果玛丽沃跑到最左边，让它再从右边出来
100 SPRITE 0，X，120	让玛丽沃出现在坐标为(X，120)的地方
110 GOTO 50	让程序回到第50句重新开始

请读者输入该程序，检查无误后，运行一下，并按动1号操纵器的左、右方向键，看看玛丽沃的运动情况是否符合要求。

上面的程序例子，只能操纵玛丽沃左右行走，而不能上下行走，这是因为第100句SPRITE 0，X，120中的水平坐标X是可变的，而竖直坐标Y不可变，固定为120，要想使玛丽沃上下左右都能行走，则必须设法也改变垂直坐标Y，请在例3—14中加入以下几行：

45 Y=120	让Y坐标的初始值为120
72 IF S=4 THEN Y=Y+1	若按向下方向键，则Y坐标加1
74 IF S=8 THEN Y=Y-1	若按向上方向键，则Y坐标减1
92 IF Y>240 THEN Y=10	若Y值大于240，即玛丽沃走到最下端，则让Y=10，即让玛丽沃再从上方出来
94 IF Y<10 THEN Y=240	同上

并修改第100句为：

```
100 SPRITE 0，X，Y
```

这时再运行该程序，则可操纵玛丽沃四方行走了。

三、使卡通的运动具有动画效果

你已学会了如何调出不同卡通图案和如何通过操纵器控制卡通的运动。现在我们来学习更多的技巧。请你复习一下前面学习的内容，在确定你掌握之后，再来学习下面的内容。

在以上例子中，卡通的运动具有很多不足之处，例如，玛丽沃运动时只有一种姿势，没有动画效果，而且不论朝哪个方向运动，玛丽沃的脸总是朝左等等。其原因是我们只定义了一种姿势，而要使卡通的运动具有动画效果，至少应有多个姿势交替出现：

立正、走、立正、走……

我们称这种效果为一个动作，当然一个动作也可由三个以上姿势组成，比如玛丽沃走的动作可以由“走 1”和“走 2”两个姿势组成，也可由“走 1”、“走 2”和“走 3”三个姿势组成。下面我们以企鹅为例，说明如何产生动画效果。

1. 姿势变换

如上所述，要产生动画效果，就要定义多个姿势。这里我们定义两种姿势，“走 1”和“走 2”。方法是：

“向左走 1”：DEF SPRITE 0，（0，1，0，0，0）＝CHR＄（96）＋CHR＄（97）＋CHR＄（98）＋CHR＄（99）

“向左走 2”：DEF SPRITE 1，（0，1，0，0，0）＝CHR＄（100）＋CHR＄（101）＋CHR＄（102）＋CHR＄（103）

企鹅的上述两种姿势都是面朝左的，即当它们组成走的动作时，只能代表向左走。没有向右走的企鹅怎么办呢？只要把企鹅图案左右翻转一下就可以了。回忆一下 DEF SPRITE 命

令，它的第 5 个参数是指明卡通图案是否左右翻转的。因此，我们可以这样定义：

向右走 1：DEF SPRITE 2,(0，1，0，1，0）＝CHR $（97）+CHR $（96）+CHR $（99）+CHR $（98）

读者可能注意到了，上边等式的右边 4 个参数顺序不是 96，97，98，99，而是 97，96，99，98。这是因为每个卡通图案都是由 4 块组成的，每个数字对应一块，虽然 DEF SPRITE 的第 5 个参数指明要作左右翻转，但只是指明每一小块翻转。因此要使整个图形翻转，还必须将左边的块调到右边来，右边的块调到左边去。同理，向右走 2 的姿势应该这样定义：

向右走 2：DEF SPRITE 3（0，1，0，1，0）＝CHR $（101）+CHR $（100）+CHR $（103）+CHR $（102）

上面定义了 4 个姿势，实际上是 4 个卡通，分别是 0 号，1 号，2 号和 3 号。可见，0 号与 1 号卡通交替出现，产生了企鹅向右走的动画效果。而 2 号与 3 号卡通交替出现，可产生企鹅向左走的动画效果。

【例 3－15】 编一段程序，让企鹅从屏幕的最右边走到最左边，然后再从最左边走到最右边。

程序：

```
10 CLS
20 SPRITE ON                                      打开卡通面
30 DEF SPRITE 0,(0,1,0,0,0)=CHR $(96)             定义 0 号卡通为
                                                  企鹅向左走 1
  +CHR $(97)+CHR $(98)+CHR $(99)
```

40 DEF SPRITE 1,(0,1,0,0,0)=CHR $ (100) +CHR $ (101)+CHR $ (102)+CHR $ (103)	定义 1 号卡通为企鹅向左走 2
50 DEF SPRITE 2,(0,1,0,1,0)=CHR $ (97) +CHR $ (96)+CHR $ (99)+CHR $ (98)	定义 2 号卡通为企鹅向右走 1
60 DEF SPRITE 3(0,1,0,1,0)=CHR $ (101) +CHR $ (100)+CHR $ (103)+CHR $ (102)	定义 3 号卡通为企鹅向右走 2
70 C=0 :D=1	变量 C,D 代表卡通号
80 FOR X=250 TO 10 STEP -1	
90 SPRITE C,X,120	让 C 号卡通出现在(X,120)位置,C 的值可以变化
95 SPRITE D	让 D 号卡通消失,否则会有两个卡通
100 PAUSE 10	不让卡通走太快
110 SWAP C,D	交换 C 和 D 的值
120 NEXT	
125 SPRITE D	让 D 号卡通消失,否则会有两个卡通
130 C=2 :D=3	
140 FOR X=10 TO 250	
150 SPRITE C,X,120	
155 SPRITE D	
160 PAUSE 10	
170 SWAP C,D	

```
180 NEXT
190 END
```

110，170 两行中用到 SWAP 语句，它的作用是把后面两个变量的值交换。如果原来 C＝0，D＝1，则在执行 SWAP C，D 之后，C 的值为 1，D 的值为 0。

现在请运行一下这个程序，看看结果是否像我们预料的一样。

如果将例 3－15 改动一下，则可以使企鹅上下走动。

2．跳跃

上面我们只能让卡通在上、下、左、右四个方向活动，而要产生更为逼真生动的效果，还要使用 DEF MOVE 和 MOVE 这一对语句。

（1）DEF MOVE 为定义卡通动作的语句

格式为 DEF MOVE(N)＝SPRITE(A,B,C,D,E,F)

各参数的含义如下：

N：卡通动作号（0～7）。

A：卡通种类（0～15）即指玛丽沃，丽莎等 16 种。

16 种卡通种类代码为

0 玛丽沃	4 企鹅	8 太空杀手卫星	12 激光
1 丽莎	5 火车	9 太空船	13 乌龟
2 苍蝇	6 车	10 爆炸	14 螃蟹
3 飞鱼	7 太空站	11 妖怪	16 鸟

B：卡通运动方向（0～8）。各方向的代码如图 3－6 所示。

C：卡通运动速度（1－255），C＝1 时最快，1 秒钟运动 60 个点；C＝255 时最慢，255 秒运动 60 个点。

D：位移量（1－255），指卡通运动距离。整个运动范围为 2×255 个点，每次移动二个点，当 D＝255 时，卡通可在屏幕出

规二次。D不能为零。

E：优先度（0—1），表示卡通在0面还是1面显示。卡通0面在背景之前，1面在背景之后，背景若为稀疏的网状物，仍隐约可见卡通。

F：配色代码（0—3）。

如 DEF MOVE（0）=SPRITE（10，1，1，100，0，0）

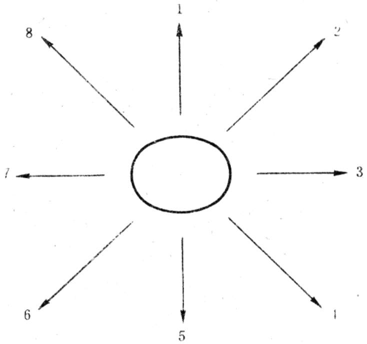


图 3—6 卡通方向代码

可解释为：规定0号动作为爆炸，以每秒60点的速度向上移动200点。在背景前显示。可以使用于打斗类游戏。

又如 DEF MOVE（4）=SPRITE（11，2，2，35，11）

可解释为：4号动作是妖怪，以每秒30点的速度向右上运动两点，在背景后显示。

（2）MOVE 启动语句

格式为 MOVE N0，N1，N2，N3，N4，N5，N6，N7

N0—N7 为启动的动作编号，对应 DEF MOVE（N）中的 N。该语句可允许一次启动1—8个动作。

例如 MOVE 0 启动0号动作；MOVE 0，1，3 启动0号、1号和3号动作。

（3）POSITION 定位语句

格式为 POSITION N，X，Y

N：DEF MOVE 语句中的动作编号（0—7）。

该语句的作用是规定一个动作的启动位置，因为 DEF MOVE 对动作进行了各参数的规定，用 MOVE 语句就可以启

动这个动作，但尚未通知电脑，这个动作从何处开始，那就要用 POSITION 语句来规定起始处的 X，Y 坐标。

例如 POSITION 0，150，200
规定了 0 号动作从 X＝150，Y＝200 坐标位置开始启动。

有了以上三个语句，我们在程序中可以产生出多种多样的卡通动作了。

【例 3－16】 编程序逐个显示 16 个卡通并在屏幕上行走。

程序 A

```
10 CLS ：SPRITE ON
20 FOR I=0 TO 15
30 DEF MOVE(0)=SPRITE (I，3，1，150，0，0)
40 POSITION 0，10，200
50 MOVE 0 ：PAUSE
60 NEXT
70 END
```

将上述程序输入电脑学习机，按下 RUN<CR>后可以看到 0 号卡通（玛丽沃）出现在屏幕左端，并从坐标 10，200 处向右行走，直到走出屏幕，然后又从左端返回行走，然后停下来。击下任意键后 1 号卡通（丽莎）又重复上述过程，直到 16 种卡通全部重复一遍。程序结束。

我们在 30 行语句中规定了 0 号动作为：I 号卡通，向右，按每秒 60 个点的速度移动 150×2 个点。在 40 行语句中规定 0 号动作起始位置为 10，200。这样在 50 行启动 0 号动作后，0 号卡通完成全部位移，遇到 PAUSE 语句，卡通停下，等待用户键盘输入后才继续执行。此时 I 比刚才增加 1，所以第 30 行中的卡通定义又变换了。

如果我们把程序稍加修改，可以一次“检阅”两个卡通，直到全部卡通“检阅”完毕。

程序 B

```
10 CLS : SPRITE ON
20 FOR I=0 TO 15 STEP 2
30 DEF MOVE (0) =SPRITE (I, 3, 1, 150, 0, 0)
40 DEF MOVE (1) =SPRITE (I+1, 3, 1, 150, 0, 0)
50 POSITION 0, 10, 200
60 POSITION 1, 10, 100
70 MOVE 0, 1 : PAUSE
80 NEXT
90 END
```

关于游戏编程还有许多语句可用，像我们上面提及的配色、卡通1面和背景面编程等，都未提及，要真正编出实用性和趣味性很强的游戏程序，不是这一本书可以用廖廖数语介绍完的。我们在此只想给读者一个粗浅的概念，让读者明白，如果想学，这并不难。对于有兴趣作更详细了解的读者，可参考下列读物：

（1）各电脑游戏机的随机说明书；

（2）于春、张新莲编著的《F BASIC 语言与编程技巧》（电子工业出版社，1993 年 4 月出版）。

第四章　复杂程序编写

在第三章的程序练习基础上，我们可以有能力编写出一些较为复杂的程序。实际上，所谓复杂，也是由许多简单的部分组合而成，所以把一个复杂程序解剖一下，分成若干个功能组成的子块，分析每个子块，也就不复杂了。

本章我们将第三章已经分析、使用过的程序加以进一步提高，使之更具有实用性，更加生动，从而进一步提高编写程序的能力。

4.1　将不同的功能块结合

前面我们分别介绍了电脑计算、辅助教学等具有各种功能的程序。我们可以把这些独立的部分都当成"子块"，当把子块结合起来时，程序就有了更多的功能了。

【例 4－1】　将例 3－3 推算星期程序加入游戏画面，待结果（星期几）推算出来后，用卡通来表示，并发声提示用户。

我们可以这样来设计画面。用 0－6 号卡通分别表示星期日、星期一至星期六。将这些卡通从上到下排列在屏幕左侧，待算出星期数后，发出一个嘟嘟声，并令这个相应的卡通在屏幕上行走。如算出某日是星期四，则令 4 号卡通（企鹅）行走，其它不动。

要达到这个目的，我们把例 3－3 和例 3－16 稍加修改，并

"拼"到一起就可以了。

程序（整数 BASIC）：

```
10  DIM A (12)
20  FOR I=1 TO 12
30  READ A (I)
40  NEXT
50  INPUT " NIAN, YUE, RE"; N, Y, R
60  M=M-M/100*100
70  W=M+M/4+A (Y) +R
80  S=W-W/7*7: CLS
90  PRINT " QINGQI="; S
100  DATA 0, 3, 3, 6, 1, 4, 6, 2, 5, 0, 3, 5
110  SPRITE ON
120  FOR I=0 TO 6
130  DEF MOVE (I) =SPRITE (I, 3, 1, 10, 0, 0)
140  POSITION I, 10, 40+I*30
150  MOVE
160  NEXT
170  DEF MOVE (S) =SPRITE (S, 3, 1, 150, 0, 0)
180  BEEP: BEEP: BEEP
190  POSITION S, 10, 20+S*30
200  MOVE S
210  END
```

从上述程序我们可以看出，从 10～100 行与例 3－3 中程序并无变化。在分析下半段程序之前，我们先认识一下第 180 行的 BEEP 语句。

BEEP 语句是简单的声音语句，能使扬声器发出"嘟"的蜂

鸣声，嘟声十分短促，为了维持一定发声长度，可以用多个 BEEP 语句，如：

```
10 BEEP ：BEEP ：BEEP
```

或使用循环语句：

```
10 FOR I=1 TO 5 ：BEEP ：NEXT
```

通过 80 行以前的计算，我们得知待查日子为星期 S，120—160 行的作用就是在屏幕左端，从上到下列出 7 个卡通来。当然可以逐个定义，逐个出现。为了提高效率，我们使用了循环语句。当 I=0 时，130 行定义动作 0 为：让卡通 0，向右以每秒 60 点的速度移动 2×150 个点，然后停下。140 行则规定起始点为 X=10，Y=40+I×30 的位置。I=0 时 Y=40，下一个卡通的位置 X 不变，因 I=1，Y=70，最后一个卡通的位置（I=6），Y=220，没有超出最大允许的 Y 值（239）。当 I 从 0 到 7 循环完毕时，7 个动作和它们的起始位置都完成了，并由 150 行语句把相应的卡通调到屏幕上。在这里我们把 0 号卡通的位置设计靠下一点，这样不与 90 行打印出的星期几信息互相重叠。

下面要做的就是发声提示，并令 S 号卡通从它的起始位置开始行走一段距离以表示该日子是星期 S。180 行用三个 BEEP 语句作发声。170 和 190 把 S 号动作再行定义一下，200 行则启动 S 号动作，从而完成全部程序。

有几点请读者注意：

(1) 本程序尽管 210 行处有 END 语句，但是程序运行结束后，显示并未关闭。如果还有第二个日子需要查询，要键入关

闭语句，并重新启动程序。操作如下：

```
SPRITE OFF<CR>
RUN<CR>
```

（2）如果我们在210行前加一个关闭显示语句：

```
205  SPRITE OFF
```

整个卡通显示一闪而过，不待完成全部卡通动作就将显示关闭。

如果要将程序设计成可多次询问的形式，可将程序尾部改写如下：

```
200  MOVE S：PAUSE 100
210  SPRITE OFF
220  GOTO 50
230  END
```

4.2　在原有程序中增加功能

我们在读懂、理解一个程序中各变量的含义和程序的结构之后，就可以在这个程序的基础上增加一些功能。或者把程序修改得更具有实用价值，使用更方便。用“行话”来讲，就是说程序与用户的“界面”更为友好。如我们第二章中介绍的学习卡，与用户有两种界面，一种是菜单式，一种是询问式，两者相比，显然菜单式比询问式“友好”得多，它无需在用户进入某一功能前多次击键，去回答电脑的许多问题。

【例 4—2】 将例 3—4 乘法练习的程序加以改编，使它的功能更强，能做“升级”和“退级”处理。

程序设计如下：

如果用 J 表示出题的级别，J 从 1 开始，相当于游戏中过关的“关数”。则：

（1）答题正确率≥80%时 J+1→J；

（2）答题正确率≤50%时 J—1→J，若 J 为 1，则不减 1，仍为 1；

（3）答题正确率在 50%至 80%之间时，J 不变；

（4）例 3—4 中打印错题等功能维持不变；

（5）第一级为一位数乘以 1—10 练习，第二级为一位数乘以 1—20 练习等等。

程序：

10 CLS	
20 DIM A（10），B（10），C（10），R（10）	
30 J=1：PRINT " 第一级"	J 为级数，第一级开始
40 T=0：L=0	T：对题数；L：错题数
50 FOR M=1 TO 10	
60 A（M）=RND（10）	A（M）：被乘数，总是一位数
70 B（M）=RND（J*10）	B（M）：乘数，随 J 增大而增大
80 X=A（M）*B（M）	X：正确答案
90 PRINT A(M);"X";B(M);"=";	列出算式，提示用户作答

```
100  INPUT C(M)                              C(M)为用户答案
110  IF C(M)=X THEN GOTO 160                 答对,转向 160 行
120  PRINT "错了!"                            否则作出错处理
130  R(L)=M                                  R(L)计录错题
                                             序号
140  L=L+1                                   错题数加 1
150  GOTO 180
160  T=T+1                                   对题数加 1
170  PRINT "很好,请继续"
180  NEXT
190  PRINT "正确率=" ;T * 10 ;" %"
200  IF L< >1 THEN PRINT "错题如下:"
210  FOR M=1 TO L-1
220  IF L=1 THEN M=L-1: GOTO 250          如无错,不打印
230  Y=R(M)
240  PRINT A(Y); "X"; B(Y);"="; C(Y);循环打印所有错题
     " ?"
250  NEXT
260  IF T>=8 THEN J=J+1                      对八题以上,升级
270  IF T<=5 AND T< >1 THEN J=J+1  对五题以下,降级
280  PRINT:PRINT:PRINT                       "下一轮为第";J;
                                             "级"
290  GOTO 40
300  END
```

从上面程序可以看出,与例 3—4 相比,这个程序的改动并不大,但其功能却比原来大大增强了。用这种方法对程序加以提高对初学者来说虽然慢一些,但其难度就大大减小了。等到熟练到一定程度,当然也可以一次就写出较完整的程序。

这里关键是增加了一个控制级数变量J。

至250行为止，程序已完成出题、给出本轮报告书和打印本轮全部错题的任务，下面要判断一下根据本轮练习结果，应该升级、降级或维持原级别。

第260行如满足T≥8，则说明本轮练习正确率为80%以上，就执行J=J+1语句，对于这时的T，肯定不会满足270行的条件，所以不可能执行270行中J=J－1语句。

如果T＜8，则260行中条件不满足，进入270行的判断。270行的条件是T≤5且T≠1，如果T≤5，说明本轮练习不理想，应退一级，即J=J－1，但如果此时J=1，则再减去1，就为0了，所以对于J=1的情况，不能做J－1运算。只有在两个条件同时满足时，才去执行J=J－1。

如果某个T对于260和270的条件均不满足，说明这时的T有几种可能：要么T在6－7之间，要么T=1，对于这两种可能，J维持原来的值，不作变化．即下一轮练习与上一轮的级别相同。

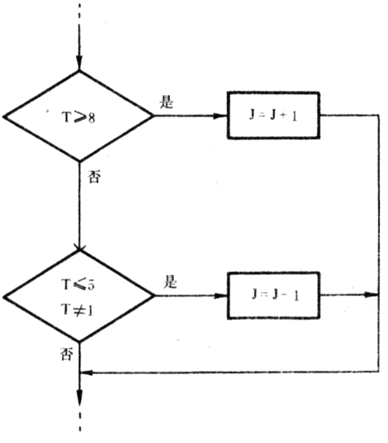


图 4—1 判别的框图表示

对于上面的种种可能，我们在写程序时，可以用一个图来清晰明了地表示。这种图称为“框图”或“流程图”。一般用菱形框表示判别，判别框有一个入口和两个出口，出口分别为条件成立（是）和不成立（不是），方框用来表示各种处理。

4.3 用框图描述功能

【例 4—3】 将例 3—5 英文字母击键练习加以改进。可用声音提示击键正确与否；同时每轮练习结束可给出击错键的键名。

本练习既要把各个子块结合起来，又要对某一部分加以提高。我们可以根据题意，画出整个程序的框图，变量名仍采用例 3—5 所用，W 为正确率，L 为难度。框图如图 4—2 所示。

图 4—2 比图 4—1 框图多了两种框，一种是平行四边形框，在流程图中用来表示输入或输出操作。另一种是两端带圆弧的框，来表示开始或结束。

让我们按步就班，把各个框逐个完成，然后，像搭积木块一样把它们拼接起来，组成完整的程序。

（1）发声 A 和发声 B

有能力的读者可以发挥自己的想象，编出两段不同的乐曲作为击键的反应。为了说明问题方便起见，我们用 BEEP 语句作为发声 A，发声 B 用下面曲子：

7. 6　7. 6　7. 6

```
PLAY " T2Y2M0V15O3"
PLAY " B4A1B4A1B4A1"
```

（2）记录击键错误的字符

我们需要定义两个字符型数组，如 WA＄和 RA＄，每击错

一次，把击错的键名和应击的键名都分别存入 WA＄和 RA＄数组中，最大个数都规定为20。在出报告单时，将它们列出，以

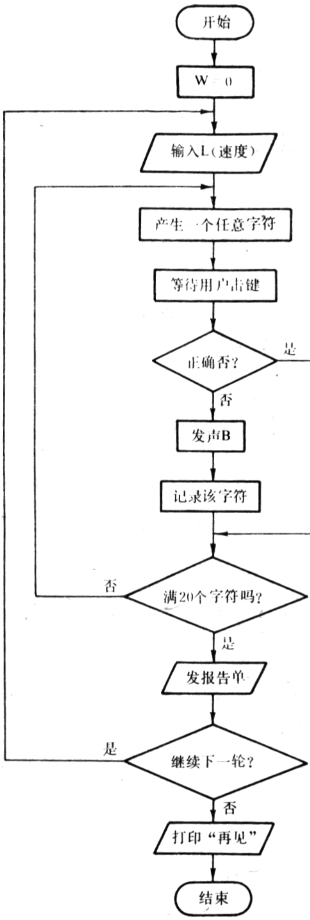


图 4—2　例 4—3 框图

便用户分析自己的击键过程，哪些键容易击成另外的哪些键，找出原因，加以改进。

本程序用 WA＄数组装错打的键名，RA＄数组装电脑产生的键名。可以采用例 3－4 和例 4－2 中记录错误练习题的方法来记录错击的键。读者也可试试自己编制这个程序。

（3）发报告单

有了 RA＄(20)和 WA＄(20)两个数组，发报告单仅仅是设计打印语句的问题了。

程序：

```
10  DIM  WA$(20),            WA$：应打字符，RA$：打错
    RA$(20)                  字符
20    CLS
30    INPUT "请选择速度";L    L 为速度指标，请用户选择
40    V=0:W=0                V：正确击键数，W：错误击
                             键数
50    FOR R=1 TO 20          一轮 20 个字符
60    A=RND(89)
70    IF A<65 THEN GOTO 6    非字母，重选
80    X=RND(27): CLS         X 为字符下落处的 X 坐标
90    FOR Y=1 TO 19          Y 为字符下落处的 Y 坐标
100    LOCATE X,Y:
        PRINT CHR$(A)        从 X,Y 处开始下落
110    LOCATE X,
        Y-1:PRINT " "        落下一个位后，消除上一个字符
120    FOR P=1 TO L*10       P 循环控制发字符速度
130    B$=INKEY$             扫描键盘，把击下的键送 B$
140    IF ASC(B$)=0
        THEN GOTO 210        如用户还没击键，P 继续循环
```

```
150    IF CHR$(A)=B$
        THEN BEEP:BEEP       如击键正确,作相应处理
        :P=L*10:Y=19:V=V+1:GOTO 160
160    PLAY "T2Y2M0V15"
170    PLAY "O3B4A1B4A1B4A1"
180    W=W+1
190    RA$(W)=CHR$(W): WA$(W)=B$
200    Y=19: P=L*10
210    NEXT
220    NEXT
230    IF ASC(B$)=0
        THEN W=W+1 :         如字符落地仍未击键,作相应处理
        RA$(W)=CHR$(A)
240    NEXT
250    CLS
260    LOCATE 10,0: PRINT "正确率为";报告正确率
        V*5;"%"
270    IF W=0 THEN GOTO 320      如无击错,不打印错字符表
280    PRINT "屏幕字符","键入字符"
290    FOR R=1 TO W
300    PRINT RA$(R),WA$(R):PRINT
310    NEXT
320    LOCATE 5,12 :PRINT "继续下一轮?(Y/N)"
330    INPUT P$
340    IF P$="Y" THEN GOTO 20
350    LOCATE 10,18: PRINT "再见!"
360    END
```

先说明一下 140 行和 230 行中 ASC(B$)函数的作用。

我们前面已经了解了 INKEY $ 的作用,INKEY $ 执行时会扫描键盘,把键盘上按下的键名送到 INKEY $,如果没有任何键按下,INKEY $ 得到一个空字符,此空字符的码转换成数字即为 0。

ASC(B $)的作用是把 B $ 的 ASC II 码转换成数值。而从 130 行知 B $ 里装的是按下的键名,如果在 INKEY $ 执行期间没有任何键被按下,则可得知 ASC(B $)=0。从而我们可以用 ASC(B $)=0 的成立与否判别出有无键被按下。

下面对程序中若干问题加以说明:

(1) 120—210 行的循环,循环变量为 P,用以控制电脑发字符的速度,循环次数为 L×10 次,所以用户在 30 行输入 L 的值,控制发字符速度。

130 行 B $ =INKEY $ 就是通过 P 的循环,监视有无键被按下,一旦按下时,就把键名送入 B $,如没有按下,则 140 行中 ASC (B $) =0 肯定成立,就跳到 210 行,使 P 增加 1,继续监视键盘。

在某个 P 值下,如果键盘被按下,那么只有两种可能:按下键与电脑发生字符 CHR $ (A) 一致,或者不一致。150 行作两者一致的处理:发声,正确率计数器 V 增加 1,P=L×10 和 Y=19 两个语句是告诉电脑不必再作 P 的循环了,即不必再继续监视键盘了,同时也不必将击对的字符继续下落了。当 Y=19 时,程序会自动退出 90 行的循环。Y 循环用于控制屏幕字符不断下落。

反之,如果不满足 150 行中的条件,那一定是击错了键,程序不会执行 150 行 THEN 后面的部分,而转向 160 行开始执行。从 160—200 是击键错误的处理。这时把错键计数器 W 增加 1,把应击字符和错击字符分别装入 RA $ 和 WA $ 两个数组。

然后退出 P、Y 两个循环。

(2) 230 行处理另一种可能性，就是 P 循环完毕，Y 也循环完毕(字符落到屏幕底部了)，这时由于用户击键不熟练，仍未击下任何键，这种情况应记入击键错误类，并且在出报告书时报出应击而实际轮空的键名，提醒用户多加练习，所以在 230 行中作相应处理。

4.4 屏幕作图

【例 4—4】 将例 3—6 计算人体节律的程序加以改进，使具有下列功能：

(1) 能用图形显示出待查日在一周期中的位置；

(2) 出生年份从 1900 年起，待查年上限不加限制。

第(1)个功能主要侧重于屏幕图显示的处理。我们在例 3—6 中已计算出了一个量 Q，现在只要将一个周期显示出来，并将待查日的 Q 在周期中的位置标出就能实现。

由于 BASIC 程序显示使用的是背景面(见第三章图 3—4)，而背景面的显示比较粗糙，在 X 方向只有 27 个点的位置，Y 方向 23 个点(见图 3—1)，所以不可能作出很细微的曲线或图形。本例只是作为图形曲线的示例，让读者得到一个作图的概念。

由于智力周期为 33 天，情绪周期为 28 天，如果在 X 方向依每天占一个点来设计图形，就不能在屏幕上显示一个完整的周期，所以采取 2 天占一个点来作图。在 Y 方向采取最大值为周期数的 1/4，最小值也为周期数的 1/4，整个曲线的垂直方向点数就等于周期数的 1/2，例如对智力周期为 16 点(舍去半

天)，对情绪周期是 14 点，均不超出 Y 方向的最大点数。

第(2)个功能的实现可参考例 3—3 由年、月、日推算星期的办法来解决。因为当出生日和待查日之间距离很大时，计算出来的天数会很大。在 BASIC 中，整数的存放和表示范围是有限制的，一般不超过绝对值 32767。所以为防止整数过大，在计算天数的过程中，设法把正在计算的节律的整周期天数扣除，不加到总天数里去，而只将余数部分加入，这并不影响计算的准确性。

程序

```
10   DIM A(3),B$(3)                      A 数组存周期数,B$ 数组存
                                          打印字符
20   A(1)=23:A(2)=33:A(3)=28
30   B$(1)="健康":B$(2)="智力":
     B$(3)="情绪"
40   PRINT "请输入出生年,月,日"
50   INPUT N1,Y1,R1
60   PRINT "请输入待测年,月,日"
70   INPUT N2,Y2,R2
80   N=N2-N1:
     Y=Y2-Y1:R=R2-R1                     计算出年差,月差,日差
90   P=0                                  P 为闰年数
100 FOR S=1900 TO
     N2 STEP 4                            每 4 年一个闰年
110 IF S>=N1 AND
     S<=N2 THEN P=P+1                    计算出生年到待查年的闰
                                          年数 P
120 NEXT
130 FOR I=1 TO 3                         计算三个周期的 Q
140 M=365-365/A(I)
```

```
    *A(I)*N                              扣除整周期
150 M=M+P+30*
    Y+Y/2+R                              计算天数
160 Q=M-(M/A(I))*A(I)                    算出第I个周期的状态
170 CLS
180 PRINT B$ (I)
190 FOR E=1 TO 20 STEP 2
200 LOCATE Q/2,                          在待查日处打一列*
    E: PRINT " *"
210 NEXT
220 Y=10
230 FOR F=0 TO A(I)
240 IF F-F/2=F/2
    THEN GOTO 280                        如为奇数日，不作图
250 IF F< A (I) /8 OR
    F>=3*A (I) /8                        上升期，Y向上
     THEN Y=Y-1
260 IF F>= A (I) /8 AND
    F<3*A (I) /8                         下降期，Y向下
     THEN Y=Y+1
270 LOCATE F/2, Y: PRINT "."
280 NEXT
290 PRINT " 击任意键继续"
300 PAUSE: CLS                           等待用户击键，计算下一个
                                         周期
310 NEXT
320 END
```

本程序大部分内容与例 3—6 相同，读者不难读懂。

作图的设计采用逐个周期处理的方法。I=1时计算健康周期，并作图，然后等待用户击键，进入I=2，作智力周期的计算和作图，直到三个周期全部处理完毕，退出程序运行。

几点说明：

(1) 由于Y方向的起点在左上角，即最上面坐标为零，所以曲线越高则Y值越小。

250和260行判断F在一个周期的上升期（前1/4周期和末1/4周期）还是下降期（中间1/2周期），如为上升期Y每次减1，坐标位置上升；若为下降期，Y每次加1，坐标位置下降。

由于把曲线压缩了一半，所以用除以8来判断点在周期中位置。

(2) 本程序作出的曲线并非正弦曲线，而是由直线段构成的。这是因为在整数BASIC下要作正弦曲线是很困难的。但我们可以从中看出待测日在周期中的相对位置了，也可得知待测日前后一段时间的变化趋势。

4.5 内存占用和执行时间

【例4—5】 对程序例3—7加以改进，使它尽量少占用内存并能提高运行速度。

首先我们来观察例3—7提出的思考问题（1），为什么我们在程序例3—7中不使用一个循环语句把所有数据读完。

大家可以看出有两个原因。第一，由于A1—B3六个数组的大小不一样，使用同一循环语句时，其循环上限不好确定。第二，是为了DATA语句中列数据的方便，在例3—7中，我们写DATA语句数据的顺序是按照男7岁、男8岁、男9岁、女7岁、

女8岁、女9岁来填入的，填完一组，接着填下一组。在执行READ语句时，也按顺序读入相应的数组。所以针对上述原因，完全有可能改成一个循环语句读完所有数组。

首先把所有数组按最大元素个数36来定义，凡数据不够36个的用零来补足。这样在程序110—240行不变的情况下，可按下面改写：

程序A：

```
10   DIM A1 (36), A2 (36), A3 (36)
20   DIM B1 (36), B2 (36), B3 (36)
30   A$=" 成年后身高预测为:"
40   FOR I=1 TO 36
50   READ A1 (I)
60   READ A2 (I)
70   READ A3 (I)
80   READ B1 (I)
90   READ B2 (I)
100 READ B3 (I)
110 NEXT
    :
    :
220 END
230 DATA 157, 152, 146, 151, 145, 140, 159, 154, 147, 153,
    147, 141
240 DATA 160, 155, 149, 155, 149, 143, 162, 157, 151, 156,
    150, 144
250 DATA 164, 159, 153, 158, 152, 146, 166, 161, 154, 160,
    154, 148
260 DATA 168, 163, 156, 162, 156, 149, 170, 164, 158, 163,
```

```
     157，151
270 DATA 172，166，159，165，159，152，173，168，161，167，
     161，154
280 DATA 175，170，163，169，162，156，177，171，165，170，
     164，157
290 DATA 179，173，166，172，166，159，181，175，168，174，
     167，161
300 DATA 183，177，170，176，169，162，184，179，171，177，
     171，164
310 DATA 186，180，173，181，174，167，188，182，175，183，
     176，169
320 DATA 190，184，177，184，178，170，192，186，178，186，
     179，172
330 DATA 194，188，180，188，181，174，195，189，182，190，
     183，175
340 DATA 197，191，183，191，184，177，199，193，185，193，
     186，178
350 DATA 201，195，187，195，188，180，203，196，189，195，
     188，180
360 DATA 205，198，190，0，189，182，0，200，192，0，191，183
370 DATA 0，202，194，0，193，185，0，204，195，0，194，187
380 DATA 0，205，197，0，196，188，0，0，199，0，0，190
390 DATA 0，0，201，0，0，191，0，0，202，0，0，193
400 DATA 0，0，204，0，0，195，0，0，206，0，0，0
```

特别注意，这里 DATA 语句数据排列的顺序与例 3－7 全不一样，当 I＝1 时，读循环的顺序是 A1（1）、A2（1）、（1）、B2（1）、B3（1），所以前 6 个数据是对于足长 170 毫米男 7 岁、男 8 岁、男 9 岁和足长 172 毫米的女 7 岁、女 8 岁、

9 岁数据。然后 I＝2，3.．.．.． 依次循环，则数据的排列顺序要按读语句的执行次序来排。当 I＝28 时，A（1）数据全部读完，但循环语句仍要去读 A1（29）的数据，这时必须用零来填上，否则后面所有数据的读入都会出错。同样理由，其他组数据用完时，也要用零补足。

程序经以上改写后，变得较为清晰明了，但没有省出存储空间（由于补了若干 0，还略微扩大了些），也没有节省执行时间，应设法按例 3－7 思考问题（2）的思路，使程序运行时内存只占用一个年龄段的数据，这样就大大减小了存储量，有可能在电脑学习机里把全部年龄段的数据都用上，使程序可以查阅 7 岁至 16 岁未成年人的身高预测。

实际上，要做到这一点并不难，因为前面已经讲过，READ 语句是根据 DATA 语句的“指针”来读数的，指针一开始指向 DATA 中的第一个数据，每读走一个数据，指针就向下移一个位置，再次遇到 READ 语句时，读走的就是下一个数据，于是指针再向下移，如此不断进行。在 BASIC 里设置了一个“拨指针”的语句 RESTORE，利用 RESTORE 语句，可把指针拨回到开始位置。

如下面程序：

```
10    READ      A, B, C
20    RESTORE   :    READ    D, E, F
30    RESTORE   :    READ    G, H, I
          :
          :
          :
100   DATA  3, 4, 5
```

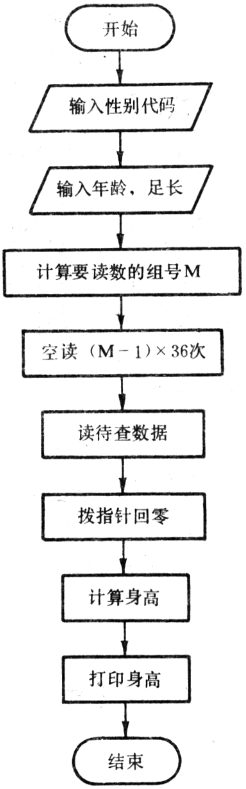


图 4—3 例 4—5 框图

虽然在DATA中只有三个数据，而READ语句共读了9个数，使变量A，B，C直到I都读到了值。原因是执行完10行后A＝3，B＝4，C＝5指针在20行RESTORE命令下返回起点DATA数据3处，再执行READ D，E，F。所以执行之后D＝3，E＝4，F＝5，同理，30行的结果使G＝3，H＝4，I＝5。

我们只用一个数组来编写例3—7程序时，可把数据按本例程序A的办法，分成6个等长的数据组，每组36个数据，不足36个的组用0补足。编成1号组至6号组，1号组为男7岁，2号组为男8岁等等。当标出要查哪一组的数据时，因为指针在1号组第一个数据处，我们前面可以做些“空读”，例如要查男8岁，则我们执行36个空读语句，把数据读到一个“无用”的变量里，等指针指到2号组第一个数时，才开始向数组里读数，读够36个数后，把指针拨回原处，等待下次读。

可用图4—3所示框图表示。

程序B：

```
10  DIM A (36)                    用一个数组装待查年龄组的数据
20  Z$ =" 成年后身高预测:"
30  INPUT " 请输入性别 (男1, 女2)"; A
```

```
40  INPUT " 请输入年龄（7—9)"；X
50  INPUT " 请输入足长（毫米)"；B
60  M=3*(A—1)+(X—6)        计算出要查第几组数据
70  IF M=1 THEN
   GOTO 110                 如为第一组，不空读
80  FOR I=1 TO (M—1)
   *36                      如不是第一组，空读M—1组数据
90  READ Y                  空读，Y并无用处
100  NEXT
110  FOR I= TO 36           读待查数据
120  READ A (I)
130  NEXT
140  R= (B—170) /2+1        由足长计算组内序号
150  IF A=2 THEN
   R=R—1                    如查女孩修改序号
160  PRINT Z$；A (R)；" 毫米"
170  END
180  DATA
250 DATA 157，159，160，162，164，166，168，170，172，173，
    175，177
260 DATA 179，181，183，184，186，188，190，192，194，195，
    197，199
270 DATA 201，203，0，0，0，0，0，0，0，0，0，0
280 DATA 152，154，155，157，159，161，163，164，166，168，
    170，171
290 DATA 173，175，177，179，180，182，184，186，188，189，
    191，193
300 DATA 195，196，0，0，0，0，0，0，0，0，0，0
310 DATA 146，147，149，151，153，154，156，158，159，161，
    163，165
```

```
320 DATA 166，168，170，171，173，175，177，178，180，182，
    183，185
330 DATA 187，189，0，0，0，0，0，0，0，0，0，0
340 DATA 151，153，155，156，158，160，162，163，165，167，
    169，170
350 DATA 172，174，176，177，179，181，183，184，186，188，
    190，191
360 DATA 193，195，0，0，0，0，0，0，0，0，0，0
370 DATA 145，147，149，150，152，154，156，157，159，161，
    162，164
380 DATA 166，167，169，171，172，174，176，178，179，181，
    183，184
390 DATA 186，188，189，191，193，194，196，0，0，0，0，0
400 DATA 140，141，153，144，146，148，149，151，152，154，
    156，157
410 DATA 159，161，162，164，165，167，169，170，172，174，
    175，177
420 DATA 178，180，182，183，185，187，188，190，191，193，
    195，0
```

程序B看上去比程序3－7简洁得多，但它运行时占据的内存空间却大大减小了，因为在这里只定义了一个A数组。运行一下试试：

```
RUN<CR>
请输入性别（男 1，女 2）?
? 1<CR>
请输入年龄（7－9）?
? 7<CR>
请输入足长（毫米）?
```

? 184<CR>

成年后身高预测为 170 厘米　与表中查出一致

OK

RUN<CR>

请输入性别（男 1，女 2）?

? 1<CR>

请输入年龄（7—9）?

? 7<CR>

请输入足长（毫米）?

? 224<CR>

成年后身高预测为 0 厘米　表中无数据，用 0 补足的

OK

RUN<CR>

请输入性别（男 1，女 2）?

? 1<CR>

请输入年龄（7—9）?

? 7<CR>

请输入足长（毫米）?

? 252<CR>

IL　ERROR　IN160　算出 R＝37，超出了定义 A（36）的范围，

OK　程序自动退出运行，指出错误。如想保护程序执行到此不退出，可在程序里加入相应语句，判断出 R＞36 时，打印"超出查阅范围，请重新输入数据"然后 GOTO 程序头部。

有几点要加以说明：

(1) 程序 B 未使用 RESTORE 语句，是因为程序每运行一

次，即每RUN＜CR＞一次，只查一组数据，再想查下一个时，要重新运行，再次运行时，指针自动回起点。若想设计成每次运行完毕不退出程序，询问用户是否继续查询，那么就要每次读完有用数据后，加一个RESTORE，在程序B里只要在130与40行之间加一个RESTORE语句即可。

（2）第60行M为待查儿童的数据组号，如查女童8岁组时，M＝3×1＋（8－6）＝5，为第5组；查男童9岁时，M＝3×0＋（9－6）＝3，为第3组。

（3）我们略去了A和X误输入时的保护语句，读者可自行加入（参考例3－7中130和155行）。现在我们已经达到了一个目的，即用很小的内存开销实现了一个很大的数据表格查阅。还有一个未解决的问题就是加快执行速度。程序B未解决速度问题，原因是从80行到100行的循环，读入所有的Y值都是无用的，当要查的数据组号很大时，会耗费许多的时间，能不能把这些时间节省下来呢？

答案是肯定的。要节省时间，就要有一个能任意拨DATA语句数据指针的办法，RESTORE只能把指针拨回起点处，这就限制了它的应用。我们设想有一个语句，可以把指针拨到所想的任何一个DATA语句行号上来，那就可以不作空读了，从而大大节省了程序的执行时间。ON RESTORE语句就可实现这一功能。

ON　RESTORE的语句格式为：

ON　A　　RESTORE　　S1，S2，S3，S4……

它的含义是根据变量A的值来拨指针，如果A＝1，指针拨到行号S1的DATA语句头部；若A＝2，则拨到行号S2的DATA语句头部，等等。这样我们就可以任意控制指针的去向了。

例如下面程序：

```
10    A=0
20    A=A+1 : IF A>3 THEN GOTO 80
30    ON A RESTORE 90, 100, 110
40    READ B, C, D, E
50    PRINT " A="; A
60    PRINT B, C, D, E
70    GOTO 20
80    END
90    DATA 11, 12, 13, 14
100   DATA 21, 22, 23, 24
110   DATA 31, 32, 33, 34
RUN<CR>
```

很显然当A=1时，读入的B，C，D，E是70行的数据，A=2时读入的是100行的数据，最后一次A=3，读入110行数据。

现在我们只要设计出变量A的变化，就可方便地实现免除空读的程序。实际上，程序B里的变量M就是一个现成的拨指针变量。

程序C：

```
10    DIM A (36)
20    Z$=" 成年后身高预测:"
30    INPUT " 请输入性别 (男1, 女2)"; A
40    INPUT " 请输入年龄 (7-9);" X
50    INPUT " 请输入足长 (毫米);" B
60    M=3* (A-1) + (X-6)
70    ON M RESTORE 200, 230, 260, 290, 320, 350
80    IF M=1 THEN GOTO 110
90    FOR I=1 TO (M-1) *36
100   READ Y
```

```
110     NEXT
120     FOR I= TO 36
130     READ A (I)
140     NEXT
150     R= (B-170) /2+1
160     IF A=2 THEN R=R-1
170     PRINT Z$; A (R); " 毫米"
180     END
200 DATA 157, 159, 160, 162, 164, 166, 168, 170, 172, 173,
    175, 177
210 DATA 179, 181, 183, 184, 186, 188, 190, 192, 194, 195,
    197, 199
220 DATA 201, 203, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
230 DATA 152, 154, 155, 157, 159, 161, 163, 164, 166, 168,
    170, 171
240 DATA 173, 175, 177, 179, 180, 182, 184, 186, 188, 189,
    191, 193
250 DATA 195, 196, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
260 DATA 146, 147, 149, 151, 153, 154, 156, 158, 159, 161,
    163, 165
270 DATA 166, 168, 170, 171, 173, 175, 177, 178, 180, 182,
    183, 185
280 DATA 187, 189, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
290 DATA 151, 153, 155, 156, 158, 160, 162, 163, 165, 167,
    169, 170
300 DATA 172, 174, 176, 177, 179, 181, 183, 184, 186, 188,
    190, 191
310 DATA 193, 195, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
320 DATA 145, 147, 149, 150, 152, 154, 156, 157, 159, 161,
    162, 164
```

```
330 DATA 166, 167, 169, 171, 172, 174, 176, 178, 179, 181,
    183, 184
340 DATA 186, 188, 189, 191, 193, 194, 196, 0, 0, 0, 0, 0
350 DATA 140, 141, 143, 144, 146, 148, 149, 151, 152, 154,
    156, 157
360 DATA 159, 161, 162, 164, 165, 167, 169, 170, 172, 174,
    175, 177
370 DATA 178, 180, 182, 183, 185, 187, 188, 190, 191, 193,
    195, 0
```

程序C看上去比原有的程序3—7简单明了，而且大大节省了内存空间和缩短了执行时间。

从本例的程序演变过程我们可以总结出以下几点：

（1）每掌握一个新的语句，对于程序的编制就多了一个有力工具，但反过来说，一种语言的语句越多，要熟悉它们，所要花费的时间也越多。作为一本编程的入门读物，本书并未介绍全部BASIC程序语句，若想编出高质量的BASIC程序，就应努力掌握BASIC的所有语句，有关这方面的读物已十分丰富，请读者选择一、二种继续学习。

（2）衡量一个程序的好坏有几个标准，首先是"可读性"较强。只有可读性强的程序才便于修改和完善。相反，结构混乱，使用大量GOTO语句的程序是难以读懂的。这一点是对于作者说的，对于用户而言，则要求它的内存占用较小，而执行时间较短。特别是对电脑学习机而言，它留给用户的内存十分有限，这些内存空间有两个用途，一方面要把你输入的程序存下来，另一方面要把程序运行过程中所有的数据保存住。所以太长的程序或者程序虽短，但使用了大量中间数据的程序，都可能超出电脑学习机的极限存储容量。

第五章　使用维护问答

本章就一些读者可能遇到的问题，作一些解释。由于问题的涉及面较广，难以分门别类，本章只选择了部分有普遍性的问题。除了涉及程序、语言和电脑学习机使用等方面外，还选择了一些简单维护的知识。

1. 什么是保留字？F BASIC 有无保留字？

一般情况下，给变量命名时可以用字母开头的字母数字串，但 BASIC 保留了一些字符串，不允许当作变量名用，称为保留字。所有的命令、语句和函数都为保留字，如 READ、BEEP、PRINT、RND 等，除此之外，F BASIC 的保留字如下：

AUTO	AND	ABS	ASC	BACKUP	BGGET
BGPUT	BGTOOL	CALL	CAN	CGSET	CGEN
CLICK	COLOR	CLEAR	CRASH	CONT	CSRL IN
CUT	DELETE	ERA	ERL	ERR	ERROR
FILTER	FIND	FRE	GAME	GOSUB	HEX $
INSTR	KEY	LEFT $	LEN	LINPUT	LOCATE
LOAD	MID $	MOD	NOT	OFF	ON
OR	PALET	PEEK	PLAT	POKE	POS
RENUM	RESUM	RESTORE	RIGHT $	SAVE	SCREEN
SCR $	SGU	SPC	STOP	STRIG	STR $
SWAP	SYSTEM	TAB	TR	VIEW	VAL
VCT	XOR	XPOS	YPOS		

上面一些字符串有许多实际上是我们尚未接触到的语句、命令和函数，凡是本书已提及的均未列入，所以大家不用专门去

记忆它，必要时查一下就行。

2. 是否各牌号机器的 F BASIC 都通用？

家用电脑学习机使用的 F BASIC 有许多版本。例如科特 FCS90 使用的是从日本 MSX BASIC 语言扩展、移植而成的，建立在整数 BASIC 基础上，而小霸王机使用的 F BASIC 是浮点 BASIC。但各版本的绝大部分语句和结构都是类似的。有些数的表示能不能接受(如小数)，最好实地到机器上试，光阅读说明书也不一定完全对号。下面几点请读者使用时特别加以注意：

(1) 常数的类型：有些不能表示小数。

(2) 有些有函数功能，有些没有或较少。

(3) 循环语句：有些循环语句要求在 NEXT 后面写上循环变量名。

例：

```
10    FOR I=1 TO 5
20    PRINT I
30    NEXT I
40    END
```

双重循环：

```
10    FOR I=1 TO 5
20    FOR L=1 TO 5
30    PRINT L
40    NEXT L
50    PRINT I
60    NEXT I
70    END
```

(4) 清屏语句：有些不认 CLS 而用 HOME 为清屏语句。

3. 各牌号的学习卡是否通用？

不通用。购学习卡或具有扩展功能的卡时，一定要针对原牌号的机器，否则无法使用。即使有时候插上另一牌号卡后，看上去绝大部分键按下后学习机都能接受，并能进入相应功能块，但总会有个别键是不兼容的。所以一般情况下，尽可能不要购买其它牌号机器上的学习卡。

4. 各牌号的磁带是否通用？

一般都是通用的，因为程序往磁带上录时，使用的信号标准是统一的，所以有通用性。从声音上判别大致有以下过程：

（1）按上录音键

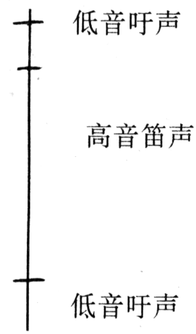
（2）按回车键开始存入 ⎤
（3）记录程序名，1 秒 ⎦ 约 10 秒

（4）等待 10 秒

（5）记录程序

（6）存入结束

（7）解除录音



5. 什么是 LOGO 语言？

LOGO 语言是一种为青少年设计的电脑语言。LOGO 语言重视教育心理学和教学法，要求系统简单、通用、有效、和谐。在计算机教学的初期一段很长时间内，程序设计离不开数字计算，而 LOGO 语言用活泼、生动的海龟图，使程序教学轻松、和谐，摆脱了令人烦恼的抽象数字计算。

用 LOGO 语言设计绘图程序，语言简洁，清晰易读。

6. 学习卡的维护应注意什么?

学习卡的维护与一般游戏卡大体相同。插卡、拔卡时一定要关闭电源后进行，否则可能损坏卡或机器。平时应注意防尘保洁，免得卡的出脚被脏物污染，引起接触不良。

一般不要将卡盒打开，如必须打开时，在将卡芯插入卡盒时不可装反，否则会引起机器不工作或元器件损坏。一般装法是线路板有元器件的面朝向卡盒的彩色标贴面。

还有一些电脑学习机的学习卡采用特殊设计，将信息用电池来维持，所以声称“具有掉电保护功能”，即关闭电源后信息并不丢失。对于这种卡，要特别注意及时更换卡内电池，一旦电池过久不换或卡长时间搁置不用，会造成电池漏液，腐蚀电池夹，并有可能危及卡内的元器件，造成永久性损坏。在此建议读者使用铁壳电池。

7. 什么是高级语言，什么是低级语言?

高级语言是用户以一种接近英语的语句来编写程序的工具，它比较易于学习、记忆和使用。如本书介绍的 BASIC 语言就是高级语言，另外 LOGO、F BASIC 也都是高级语言。低级语言是根据计算机使用二进制(我们日常使用十进制，逢 10 进 1)的特点，用一些符号和二进制代码来编程序的工具，所以不易学，也不易记忆。如汇编语言和机器语言都是低级语言。

虽然低级语言有许多缺点，但它的突出优点是速度快，节省内存。这是高级语言不能比拟的。游戏卡里的绝大部分程序都用低级语言写成。

8. 除了表 1—3 介绍的几种机型外，是否可提供更多的机型介绍？

国内生产的电脑学习机种类繁多，本书只挑选若干较为普及和有某些特色的机型作介绍，其余的，读者可根据第一、二章所述原理，通过性能介绍和说明书对它作出判断。

金字塔　9488

- 一、二级汉字，区位码、拼音码、五笔字型（包括 12000 个词组），并允许用户自行扩充一种编码输入方法，做到四码互相查找、转换，联想功能四码共享；
- 软盘驱动器接口：DOS 指令与中华学习机一致，并增加若干新指令；
- 语言：SOFTBASIC、F BASIC、LOGO 语言；
- 扩充槽：三个；
- 打印口：可外接不带中文字库的 9 针打印机，也可用 24 针打印机，27 种字形；
- 备用卡：表形码（带词组）汉卡
  直觉码（带词组）汉卡
  软盘驱动卡
  128K 内存扩充卡
  32K 内存扩充卡
  初高中各种题库
- 备选声像教学磁带：几十种。

和声 HS　2188

- 软盘驱动器口、打印机口、扩充槽口、录音机口、AV 口；
- 语言：浮点 BASIC、中文 BASIC、LOGO；
- 编码：拼音、五笔字型、区位、全角；

· 101 键标准键盘；

· 备选卡：五笔练习卡、小学数学练习卡、键盘训练卡、彩色绘画卡；

· 备选磁带：中小学辅导软件；

· 辅导报纸：定期发行。

9. 什么是“K”？

在计算机行业里通常用 K 来表示内存储器的容量大小，它是一个单位。计算机使用二进制数，所以数的大小都用 2 的幂次来表示。2 的 10 次幂等于 1024。能存储 1024 个二进制数的容量称为 1K(字节)。通常电脑学习机留给用户作 BASIC 练习的存储器容量为 8K，个别机型可达 32K。由于内存容量小，所以大的程序和数据就装不下。如果用 DIM 语句定义一、二个大的数组，电脑学习机就受不了了，会告诉你：

OM　ERR

即内存装满了，无法继续。

10. 什么是 8 位机和 16 位机？

这里的 8 位和 16 位都是指二进制数的位。8 位机处理二进制数时，每次处理长度 8 位的数，而 16 位机则可每次处理 16 位长的二进制数。可见无论从速度、精度等指标来看，16 位机都远高于 8 位机。

我们通常使用的任天堂系列游戏机，实际上都是 8 位机。现在市面上已出现 16 位的游戏机，如世嘉(SEGA)、超级任天堂、PC ENGIN 等。这些机器的画面、声音、速度都远优于 8 位机，当然价格也较高。这些机器可采用双向卷轴，立体感较强，产生

立体声，并可通过电话线与异地朋友共同游戏。

电脑学习机不能与16位游戏机相配，甚至两种游戏机卡的卡座都不通用。

11. 什么是AV口？

A即英语Audio，意思是声音；V即英语Video，意思是图像。所以AV的意思合起来指声音和图像。第二章已说明用分体式电脑学习机与游戏机配合使用时，要把游戏机的RF口与电视机天线口相连接，以这种方式工作时，游戏机的声音和图像同时从天线口进入电视机，这样使用较为方便，但声音和图像的质量较差。对于有AV口的游戏机来说，较合理的连接方式是把游戏机的Audio连到电视机的Audio，游戏机的Video连到电视机的Video，而RF线空下不用。从连接上多使用了一条线，但可提高图像和声音质量。请注意以下几点：

（1）有些电视机没有AV口，不能这样连；

（2）要用商店买来专用的AV线连，而且Audio线和Video线不能互换使用；

（3）有些游戏机机壳上标有Audio、Video字样，实际上机内没有相应电路，仅仅是装饰，所以无法使用。

12. 什么是I/O口？

I/O就是指“输入和输出”，所以也可称输出/输入口。电脑学习机需要与外部交换信息，如输入的程序需要存储，或从外部输入一个录音带上的程序等等，都是通过I/O口进行的。比较常见的I/O口有录音机输入口（LOAD），录音机输出口（SAVE），有些电脑学习机还有打印机口，软盘驱动器口等，都属于I/O范围。

13. 什么是扩充槽？

游戏机留给用户一个槽，用来插入游戏卡做游戏用。用作电脑学习机使用时，就在此槽内插入学习卡、BS 卡或从厂家买来的某种卡（如打字专家、声像卡等）。对于一体化电脑学习机来说，厂家设计机器时考虑到只给用户一个槽，使用起来不够方便。如使用打印机的用户还要插入打印卡；能配软盘驱动器的用户要插入软盘驱动器卡，所以有的机器预留了多个槽口给用户使用。如金字塔 9488 有三个扩充槽。当然若不使用那么多扩充槽，可以空着。

14. 什么是软盘驱动器？

使用录音磁带来存储程序的用户都有体会，从磁带输入和输出程序，速度非常慢，操作十分麻烦，要找出程序头部，而且录入或输出过程中容易出错，用软盘驱动器则可以免除这些不足。在存程序时，只要给机器一个程序名，电脑即自动完成一切操作。取程序时，也只要给一个程序名，即由电脑自动到软盘中去找该程序，并将它取到内存储器来。不但速度快，而且十分简便。有些电脑学习机即具有这种配置能力，如金字塔、和声等机型。软盘片类似一张密纹塑料唱片大小（还有更小的），价格与磁带不相上下，而存储容量都是“海量”，一张盘片最少可存 140K 程序，多的达 1440K 那么大，可见其存储容量多么大。

15. 什么是 DOS 操作系统？

DOS 操作系统是一种磁盘操作系统，对于使用软盘驱动器的用户，必须有 DOS 操作系统才能使用许多与磁盘有关的命令，如存文件、取文件、删除文件、复制文件等等。所谓文件，我们

可粗略地理解成程序，实际上它的含义比程序更广。

电脑学习机的 DOS 操作系统有两种设计，一般是依据苹果机（中华学习机）的操作系统来设计的，所以与 PC 机的操作系统不能通用，其命令形式也不相同。两种操作系统使用的盘片也不能互用。有些厂家为了尽量向 PC 机靠拢，在电脑学习机 DOS 中装入了 WPS 文字编辑系统，并将 WPS 下的命令设计成和 PC 机上的 WPS 软件一致。由于 WPS 软件在 PC 机上使用十分广泛，几乎是一切处理工作者必备的软件，所以在学习机上熟悉了 WPS 以后，上 PC 机操作可较快适应。

16. 什么是 BG GRAPHIC？

进入 FBASIC 后，屏幕会出现如下画面：

```
              F BASIC

      1  BASIC
      2  BG GRAPHIC
      3  END

       1, 2, 3, KEY IN! !
```

按 1 键，进入 F BASIC；按 3 键，返回主菜单，按 2 键则进入 BG GRAPHIC 功能。

BG GRAPHIC 是专用于背景绘图的，它有各种图形可供用户使用，这些图形也可以在 BASIC 程序中用专用命令调用，使整个卡通画面更加丰富生动。在 BG GRAPHIC 的背景设计完成之后，可以用 BASIC 命令将卡通画面和背景面连接在一起，构成真正的动画游戏程序。

有关 BG GRAPHIC 的绘图方法和 BASIC 程序的连接方

法，在随机说明书里可以查到。

17. 什么是“死机”？

有时候机器不知出于什么原因，处于一种状态，敲击任何键都不接受，这种状态称为死机。要摆脱死机状态，通常只有关掉机器电源，然后再重新开启电源。这样造成的损失是，有可能使已经输入了一个很长的程序不复存在。所以应尽量试试按键盘上的 Break 键，Esc 键，或试试按游戏机（分体式）上的 Reset 键，看能否摆脱死机状态。实在不能摆脱，只好采取关机的办法。在关机后，再开机时，最好间隔 10～15 秒。

18. 打印机能否打印汉字？

原则上讲打印机都可打印汉字和西文，但前提条件是必须有汉字的字库才行。有些电脑学习机本身带有字库，如金字塔等；有些则不带，需另配打印卡。另外有些打印机本身带有字库，这种打印机不仅使用方便，而且速度也特别快。

19. 什么是 CED 处理系统？

CED 是一种电子制表汉字处理系统，它除了可以编辑汉字文稿外，还可以绘制图形和表格，这是十分有用的。有些机型的 CED 系统还兼有打印功能，不需另加打印卡，就可直接与多种打印机相连打出几十种字体的文稿，如 GMT 机。所以 CED 是一种实用性较强的通用软件。

20. 国外带回来的游戏机能否作电脑学习机用？

由于游戏机是直接与电视机相连使用的，而各个国家电视机的“制式”并不一样，我国电视机采用的是 PAL 制，只有采用

PAL 制电视制式的国家，他们所使用的游戏机才有可能在我国电视机上使用。即使都是 PAL 制的，在音频方面还会有差别。不同的制式需要“改制”后才可用。对于没有电子学专业知识的读者，改制是困难的。

但如果您的电视机是属于“全制式”的，那就允许任何游戏机与它相配合。

21. 什么是卡的兼容性？

由于国内游戏机的来源十分复杂，各种生产厂家的牌号也是五花八门，所以实际上有些卡在某些机器上并不能运行，或产生死机。我们称这种机器“兼容性”差。国内游戏机基本上可分为仿任天堂系列和仿小天才系列，后者性能略好些。购买时要考察机器的兼容性，因为学习卡对机器兼容性的要求比一般游戏卡高。简单的方法是购置新的功能卡时，带上电脑学习机或游戏机当场测试，看看能否进入各个功能块，图像画面和声音是否正常。如果在没带机器的情况下购卡，则最好挑选牌号一样的机器来测试卡。

对于有卡而要选购机器的用户，则最好带卡试机。如果做不到，就用兼容性要求高的游戏节目卡来测试机器，如“成龙”、“双截龙”等。

较轻度的不兼容会产生图像抖动、扭曲、错位等现象，如角色的头发与脑袋错位、上半身与下半身错位或标题文字被割裂成两段、三段等。较严重的不兼容可能产生角色的上半身在正常位置，下半身在不同位置时隐时现，或操作时角色消失等现象，甚至在程序运行到某一部分时产生死机。

22. 怎样维护和保养软盘驱动器?

软盘和软盘驱动器是电脑配件中较为“娇贵”的部分,使用和保养都应精心,否则可能造成数据丢失或设备损坏。

软盘下方的方形窗口,是驱动器的“读写口”,所有记录和输出功能都通过这一窗口传递资料,千万不可用手摸这一部分,平时应注意不要让灰尘进入。也不要在软盘上堆压重物,以免造成形变,无法读写。对有物理损伤、受潮霉变的盘片,严禁在驱动器上使用,以免损坏磁头。

软盘驱动器也应注意防尘,不用时应用软布遮盖。驱动器不使用时,应将盘片退出,当驱动器的红色指示灯亮着时,不应向驱动器送盘和从驱动器向外取盘,否则容易损坏驱动器的读写头和软盘。

将驱动器电缆插头插入电脑学习机和驱动器口时,有些插头上有对准销,不会插反。但有些则没有对准销,可能插反。插反时,驱动器不会工作,还可能造成损坏。注意驱动器扁平电缆某一侧有一条红道,该红道是作标记用的,应在说明书指定的位置。

23. 学习卡插头或槽口不洁可能造成什么故障?

学习卡插头或槽口沾染有污物或生锈时,会使电路接触不良,常造成有声音无图像的故障,有时则会引起图像混乱、分裂,或电视屏幕出现横竖线条或斜花纹。此时可关机,将学习卡取下,查看节目卡的镀金脚是否发黑、锈斑或污染。如有污物,可用沾有无水酒精的棉花球擦去污物。如属于卡座污染,也可用酒精棉清洁卡座。如卡座簧片松弛,可用螺丝刀或镊子轻轻校正,情况严重的,可更换个别簧片或整个卡座。

24. 机器热稳定性较差，可能有哪些原因？

热稳定性较差表现在开机时图像画面和声音正常，工作一定时间后出现图像抖动、色调改变、甚至图像消失等现象。若关机后等待一段时间再开机，故障又自动排除。

出现热稳定性较差的故障，原因可能在电源部分，也可能在主机部分。由于游戏机或学习机的电源均非稳压电源，更加上有些厂家生产的电源所用变压器和整流器件并非合格品，电源外壳又小，散热较困难，很容易引起热稳定性差的故障。可用替换法缩小故障范围，另用一个电源接到电脑学习机上，若故障不再出现，可判为机器故障。

如属主机故障，很可能是由于厂家所用集成电路芯片(IC)质量不过关所致，同样型号的芯片，制造厂家可能有日本、美国、韩国、马来西亚等国家，质量和价格差异较大。一般说IC的引脚越多，表示芯片的集成度越密集，引起热稳定性差的可能性也越大。在电脑学习机主机上，有两块最大的IC，其中一块为CPU，其型号因机器牌号不同也有所异，常用的型号有6527AP、6527P、6540、或87007P。另一块为PPU，它的作用是用于作图像画面、游戏信息处理，常用型号为6528、87008P、6528、6541等。

在家庭里，可以用一块沾有无水酒精的棉花来测试是否因某个集成电路热稳定性差引起故障。方法是在开机状态下，把酒精棉挤干后轻轻擦拭某集成电路的表面，帮助其散热，如果故障消失，则说明是该芯片质差引起的故障。

排除方法是更换该芯片。作为应急处理，也可用一块大小合适的金属片，如铝片或铜片，弯曲成暖气片状，把它粘固在故障集成电路上，如故障不十分严重，加上这块散热板后，就有可能

排除。

25. 手柄是不是电脑学习机的必要部件?

一般电脑学习机不用游戏机的手柄,除非要插入游戏卡做游戏时,才会使用手柄。但也有的电脑学习机设计时,在使用录音机存取程序时,要用到手柄上的 START(开始)、SELECT(选择)等键。

26. 手柄有哪些常见故障?

手柄的常见故障有:个别按键失效、个别按键不按即动作、全部按键失效、按键功能混乱等。

由于手柄的使用极为频繁,而且使用时,可能操作力度较大,所以故障率较高。对于插拔式的手柄在其中一个故障时,可将左、右手柄互换,或暂用别的机器手柄替代,判别是主机故障还是手柄故障。当左右手柄互换时,副手柄不能当作主手柄用,而主手柄可替代副手柄用。

手柄上的按钮开关都是用导电橡胶作为接触导通媒介的,导电橡胶频繁使用后,容易老化、脱落粉末,造成各类故障。

用万用电表可检测导电橡胶的老化程度。一般地讲,导电触片任意两点的电阻应在 150Ω 以内,当测出电阻超出 200Ω 时,表示接触不良。这类故障表现现象是按键要用力按才起作用,但尚未失效。

有的导电橡胶质量较差,作为导电成分的石墨脱落,沾在橡胶之外,此时应用无水酒精棉花将其擦净,然后用万用表测试,若电阻在 150Ω 以下,仍可正常使用。

导电橡胶部分脱落,粉末可能堵塞在电路板的触点之间,造成按键按下时接触不良,或"不接即触"故障,此时只要用无水酒

精棉球将电路板上脏物擦净即可。

导电橡胶和线路板不可用油类清洁，否则会加剧橡胶老化变形。

附录 1

ASCII 码对照表

字　符	十进制代码	字　符	十进制代码	字　符	十进制代码
空格	032	＞	062	]	093
!	033	?	063	a	097
//	034	@	064	b	098
#	035	A	065	c	099
$	036	B	066	d	100
%	037	C	067	e	101
&	038	D	068	f	102
′	039	E	069	g	103
(	040	F	070	h	104
)	041	G	071	i	105
*	042	H	072	j	106
＋	043	I	073	k	107
,	044	J	074	l	108
—	045	K	075	m	109
.	046	L	076	n	110
/	047	M	077	o	111
0	048	N	078	p	112
1	049	O	079	q	113
2	050	P	080	r	114
3	051	Q	081	s	115
4	052	R	082	t	116
5	053	S	083	u	117
6	054	T	084	v	118
7	055	U	085	w	119
8	056	V	086	x	120
9	057	W	087	y	121
:	058	X	088	z	122
;	059	Y	089	{	123
＜	060	Z	090	|	124
＝	061	[	091	}	125
		\	092	～	126

附录 2

错误提示表

错误代码	出错提示	错 误 信 息	说 明
0	NF	Next without For	无 FOR 却有 NEXT
1	SN	Syntax error	语法错误
3	OD	Out of DATA	DATA 数据不足
4	IL	ILegal function	函数错
5	OV	Overflow	演算结果超出能表示的范围
6	OM	Out of Memory	内存不够
7	UL	Undefind Line Number	GOTO、GOSUB、IF 所指的行号不存在
8	SO	Subscrript Out of range	数组下标超出定义的范围
9	DD	Duplicate Defind	数组被重复定义
10	DZ	Division by Zero	用 0 作除数
11	TM	Type Mismatch	变量的类型不一致
12	ST	String too long	字符串太长，超出 31 个字符
13	FT	Fomula Too complex	式子过于复杂(如括号太多)
14	CC	Can't Continue	CONT 无法继续执行程序
17	TP	Tape read error	无法从磁带读数据
21	UP	Unprintable error	遇到无法显示打印的错误代码

说明：本表根据 F BASIC 版本号 3.10，如不同的 BASIC 和版本号不同，出错提示不尽相同。表中本书未涉及的内容从略。

附录 3

BASIC 命令、语句、函数一览表

（本书未介绍部分从略）

3—1 命令一览表

命令	功能
NEW	消除内存，不清屏幕
LIST	将内存中程序按行号列出
RUN	执行内存中程序
LOAD	从磁带中读程序
SAVE	把程序写入磁带
PRINT	打印结果或数据资料
KEYLIST	显示各功能键定义
CLS	清除屏幕，不清内存

3—2 语句一览表

语句	功能
BEEP	喇叭发声
DATA	置数据供 READ 执行
DEF MOVE(N) =SPRITE	定义卡通动作
DEF SPRITE	定义卡通
DIM	定义数组
END	程序结束
FOR - NEXT	循环
FOR - NEXT	多重循环（嵌套）

续表

语　　句	功　　能
GOTO N	无条件转移到行号 N
IF - THEN	条件语句，经判别确定执行内容
INPUT	人机对话，接受键盘输入
LOCATE	光标定位
MOVE	启动卡通动作
ONRESTORE	拨 DATA 语句数据指针
PAUSE	暂停程序执行
PLAY	乐曲演奏
PRINT	输出打印
READ	从 DATA 语句中读数据
REM	注释
RESTORE	恢复数据指针到起点
SPRITE OFF	关卡通面显示
SPRITE ON	开卡通面显示
SWAP	交换两个变量的内容

3—3　函数一览表

函　　数	功　　能
ASC(字符)	将字符转换成 ASCII 码
CHR $ (X)	求 ASCIIX 对应的字或符号
INKEY $	读取键盘输入
RND(X)	产生 X 以内的随机数
STICK(X)	读游戏手柄的方向键盘

附录 4

足长预测身高表

4—1 男子

	身				高(cm)					
足长(mm)	7	8	9	10	11	12	13	14	15	16(岁)
170	157	152	146	…	…	…	…	…	…	…
172	159	154	147	…	…	…	…	…	…	…
174	160	155	149	…	…	…	…	…	…	…
176	162	157	151	145	…	…	…	…	…	…
178	164	159	153	147	…	…	…	…	…	…
180	166	161	154	148	…	…	…	…	…	…
182	168	163	156	150	145	…	…	…	…	…
184	170	164	158	152	147	…	…	…	…	…
186	172	166	159	153	148	…	…	…	…	…
188	173	168	161	155	150	…	…	…	…	…
190	175	170	163	157	151	145	…	…	…	…
192	177	171	165	158	153	147	…	…	…	…
194	179	173	166	160	155	148	…	…	…	…
196	181	175	168	162	156	150	…	…	…	…
198	183	177	170	163	158	151	146	…	…	…
200	184	179	171	165	159	153	147	…	…	…
202	186	180	173	166	161	155	149	…	…	…
204	188	182	175	168	163	156	150	145	…	…
206	190	184	177	170	164	158	151	146	…	…
208	192	186	178	171	166	159	153	148	145	…
210	194	188	180	173	167	161	154	149	146	145
212	195	189	182	175	169	162	156	151	148	146
214	197	191	183	176	171	164	157	152	149	148
216	199	193	185	178	172	165	159	154	150	149
218	201	195	187	180	174	167	160	155	152	151
220	203	196	189	181	175	168	162	156	153	152
222	205	198	190	183	177	170	163	158	155	153
224	…	200	192	185	179	171	165	159	157	155
226	…	202	194	186	180	173	166	161	157	156
228	…	204	195	188	182	174	168	162	159	157
230	…	205	197	190	183	176	169	163	160	159
232	…	…	199	191	185	177	171	165	162	160
234	…	…	201	193	187	179	172	166	163	162

续表

足长(mm)	7	8	9	10	11	12	13	14	15	16(岁)
236	…	…	202	194	188	181	174	168	164	163
238	…	…	204	196	190	182	175	169	166	165
240	…	…	206	198	191	184	176	171	167	166
242	…	…	…	199	193	185	178	172	169	167
244	…	…	…	201	195	187	179	173	170	169
246	…	…	…	203	196	188	181	175	171	170
248	…	…	…	204	198	190	182	176	173	171
202	177	171	164	157	152	147	144	143	142	
204	179	172	165	159	154	148	146	144	143	
206	181	174	167	161	155	150	147	146	144	
208	183	176	169	162	157	151	148	147	146	
210	184	178	170	164	158	153	150	148	147	
212	186	179	172	165	160	154	151	150	149	
214	188	181	174	167	161	155	153	151	150	
216	190	183	175	168	163	157	154	153	151	
218	191	184	177	170	164	158	155	154	153	
220	193	186	178	171	166	160	157	156	154	
222	195	188	180	173	167	161	158	157	156	
224	…	189	182	175	169	163	160	158	157	
226	…	191	183	176	170	164	161	160	158	
228	…	193	185	178	172	166	163	161	160	
230	…	194	187	179	173	167	164	163	161	
232	…	196	188	181	175	168	165	164	163	
234	…	190	182	176	170	167	165	164		
236	…	…	191	184	178	171	168	167	165	
238	…	…	193	185	179	173	170	168	167	
240	…	…	195	187	181	174	171	170	168	
242	…	…	…	189	182	176	173	171	170	
244	…	…	…	190	184	177	174	172	171	
246	…	…	…	192	185	179	175	174	172	
248	…	…	…	193	187	180	177	175	174	
250	…	…	…	195	188	182	178	177	175	
252	…	…	…	…	190	183	180	178	177	
254	…	…	…	…	191	184	181	180	178	
256	…	…	…	…	193	186	183	181	179	
258	…	…	…	…	194	187	184	182	181	
260	…	…	…	…	196	189	185	184	182	
262	…	…	…	…	…	190	187	185	184	
264	…	…	…	…	…	192	188	187	185	
266	…	…	…	…	…	193	190	188	186	
268	…	…	…	…	…	195	191	189	188	

续表

足长(mm)	7	8	9	10	11	12	13	14	15	16(岁)
270	…	…	…	…	…	…	193	191	189	
272	…	…	…	…	…	…	194	192	191	
274	…	…	…	…	…	…	195	194	192	
250	…	…	…	206	199	191	184	178	174	173
252	…	…	…	…	201	193	185	179	175	174
254	…	…	…	…	202	194	187	181	177	175
256	…	…	…	…	204	196	188	182	178	177
258	…	…	…	…	206	197	190	183	180	178
260	…	…	…	…	…	199	191	185	181	180
262	…	…	…	…	…	200	193	186	182	181
264	…	…	…	…	…	202	194	188	184	182
266	…	…	…	…	…	203	196	189	185	184
268	…	…	…	…	…	205	197	190	187	185
270	…	…	…	…	…	…	199	192	188	186
272	…	…	…	…	…	…	200	193	189	188
274	…	…	…	…	…	…	201	195	191	189
276	…	…	…	…	…	…	203	196	192	191
278	…	…	…	…	…	…	204	198	194	192
280	…	…	…	…	…	…	206	199	195	193
282	…	…	…	…	…	…	…	200	197	195
284	…	…	…	…	…	…	…	202	198	196
286	…	…	…	…	…	…	…	203	199	198
288	…	…	…	…	…	…	…	205	201	199
290	…	…	…	…	…	…	…	…	202	200
292	…	…	…	…	…	…	…	…	203	202
294	…	…	…	…	…	…	…	…	205	203
296	…	…	…	…	…	…	…	…	…	204
298	…	…	…	…	…	…	…	…	…	206

4—2 女子

足长(mm)	身高(cm) 7	8	9	10	11	12	13	14	15(岁)
172	151	145	140	134	…	…	…	…	…
174	153	147	141	136	…	…	…	…	…
176	155	149	143	137	…	…	…	…	…
178	156	150	144	139	…	…	…	…	…
180	158	152	146	140	136	…	…	…	…
182	160	154	148	142	137	…	…	…	…
184	162	156	149	143	139	…	…	…	…
186	163	157	151	145	140	135	…	…	…
188	165	159	152	147	142	137	…	…	…
190	167	161	154	148	143	138	136	…	…
192	169	162	156	150	145	139	137	136	…
194	170	164	157	151	146	141	138	137	136
196	172	166	159	153	148	142	140	139	137
198	174	167	161	154	149	144	141	140	139
200	176	169	162	156	151	145	143	141	140
202	177	171	164	157	152	147	144	143	142
204	179	172	165	159	154	148	146	144	143
206	181	174	167	161	155	150	147	146	144
208	183	176	169	162	157	151	148	147	146

续表

足长(mm)	7	8	9	10	11	12	13	14	15(岁)
210	184	178	170	164	158	153	150	148	147
212	186	179	172	165	160	154	151	150	149
214	188	181	174	167	161	155	153	151	150
216	190	183	175	168	163	157	154	153	151
218	191	184	177	170	164	158	155	154	153
220	193	186	178	171	166	160	157	156	154
222	195	188	180	173	167	161	158	157	156
224	…	189	182	175	169	163	160	158	157
226	…	191	183	176	170	164	161	160	158
228	…	193	185	178	172	166	163	161	160
230	…	194	187	179	173	167	164	163	161
232	…	196	188	181	175	168	165	164	163
234	…	…	190	182	176	170	167	165	164
236	…	…	191	184	178	171	168	167	165
238	…	…	193	185	179	173	170	168	167
240	…	…	195	187	181	174	171	170	168
242	…	…	…	189	182	176	173	171	170
244	…	…	…	190	184	177	174	172	171
246	…	…	…	192	185	179	175	174	172
248	…	…	…	193	187	180	177	175	174

续表

足长(mm)	7	8	9	10	11	12	13	14	15(岁)
250	…	…	…	195	188	182	178	177	175
252	…	…	…	…	190	183	180	178	177
254	…	…	…	…	191	184	181	180	178
256	…	…	…	…	193	186	183	181	179
258	…	…	…	…	194	187	184	182	181
260	…	…	…	…	196	189	185	184	182
262	…	…	…	…	…	190	187	185	184
264	…	…	…	…	…	192	188	187	185
266	…	…	…	…	…	193	190	188	186
268	…	…	…	…	…	195	191	189	188
270	…	…	…	…	…	…	193	191	189
272	…	…	…	…	…	…	194	192	191
274	…	…	…	…	…	…	195	194	192

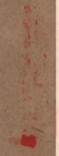
附录 5

卡通代码表

代码 10进制	说明	代码 10进制	说明	代码 10进制	说明	代码 10进制	说明	代码 10进制	说明
0	玛丽沃（走 1）	52	丽莎（跌下）	104	企鹅（正面）	156	杀手卫星（左上）	208	激光弹(1)(2)
1		53		105		157		209	
2		54		106		158		210	激光束(1)(2)
3		55		107		159		211	
4	玛丽沃（走 2）	56	苍蝇（1）	108	企鹅（背面）	160	杀手卫星（上）	212	激光刀(1)(2)
5		57		109		161		213	
6		58		110		162		214	卡通面用代码
7		59		111		163		215	
8	玛丽沃（走 3）	60	苍蝇（2）	112	火球（1）	164	太空船（左）	216	
9		61		113		165		217	音乐板用三音域(1)(2)(3)
10		62		114		166		218	
11		63		115		167		219	
12	玛丽沃（跳）	64	飞鱼（左 1）	116	168 火球（2）	A8	太空船（左上）	220	光标 1
13		65		117		169		221	光标 2
14		66		118		170		222	指示器 1
15		67		119		171		223	指示器 2
16	玛丽沃（滑）	68	飞鱼（左 2）	120	车（左 1）	172	太空船（上）	224	向上羽毛笔
17		69		121		173		225	
18		70		122		174		226	
19		71		123		175		227	
20	玛丽沃（背面）	72	飞鱼（左上 1）	124	车（左 2）	176	爆炸（开始）	228	
21		73		125		177		229	
22		74		126		178		230	
23		75		127		179		231	
24	玛丽沃（摔倒）	76	飞鱼（左上 2）	128	车（左上 1）	180	爆炸（散发）	232	向下羽毛笔
25		77		129		181		233	
26		78		130		182		234	
27		79		131		183		235	
28	丽莎（走 1）	80	飞鱼（前 1）	132	车（左上 2）	184	乌龟走（1）	236	
29		81		133		185		237	
30		82		134		186		238	
31		83		135		187		239	

续表

代码10进制	说明	代码10进制	说明	代码10进制	说明	代码10进制	说明	代码10进制	说明
32	丽莎	84	飞鱼	136	车	188		240	1
33		85		137		189	乌龟	241	2
34	（走2）	86	（前2）	138	（上1）	190	立正	242	3
35		87		139		191	（2）	243	4
36	丽莎	88	妖怪	140	车	192	螃蟹	244	
37		89		141		193		245	
38	（走3）	90	（1）	142	（上2）	194	（1）	246	COMP-
39		91		143		195		247	UTER
40	丽莎	92	妖怪	144	太空站	196	螃蟹	248	
41		93		145		197		249	
42	（跳）	94	（2）	146	（1）	198	（2）	250	
43		95		147		199		251	
44	丽莎	96	企鹅	148		200	飞鸟	252	OPER-
45		97		149	太空站	201		253	ATOR
46	（滑）	98	（走1）	150	（2）	202	（1）	254	
47		99		151			CB	255	
48	丽莎	100	企鹅	152	杀手	204	飞鸟		
49		101		153	卫星	205			
50	（背面）	102	（走2）	154		206	（2）		
51		103		155	（左）	207			



ISBN 7-115-05447-9
9 787115 054470 >
ISBN 7—115—05447—9/TP · 142
定价：6.50 元

电脑学习机两周通

人民邮电出版社